Modelling and Control of Combustion in a High Velocity Air Flame (HVAF) Thermal Spraying Process

by

Dominic Barth

A dissertation submitted in compliance with the full requirements for the

Master of Engineering in Mechatronics

In the Faculty of Engineering,

The Built Environment and Information Technology

Nelson Mandela Metropolitan University

January 2010

Promoter: Prof Igor Gorlach

Co-Promoter: Prof Gerhard Gruhler

Author’s Declaration

I Dominic Barth hereby declare that:

- The work done in this dissertation is my own.
- All other sources used or referred to have been documented and recognised.
- This dissertation has not been previously submitted in full or partial fulfilment of the requirements for an equivalent or higher qualification at any other educational institution.

_____ Date

Signature

Acknowledgements

I wish to acknowledge the contributions by the following:

My promoters Prof Igor Gorlach and Prof Gerhard Gruhler for making this whole dissertation possible, as well as for their guidance and encouragement.

Prof Theo van Niekerk and Prof Hinrich Holdack-Janssen for allowing me to use their equipment.

The department of Mechatronics, the department of Mechanical Engineering, the department of Electrical Engineering and the Automotive Components Technology Station (ACTS) for allowing me to use their facilities.

Rudolph Herselman, Terry Browning, John Fernadez and Jürgen Kranz for fruitful discussions and their support in designing and building the control system.

Frank Adlam and Rian Ehlers for their support regarding the PIC microcontroller.

Mary-Ann Snyders for her support with paperwork.

Heike Möller and Mnr. Karl du Preez for allowing me to use their laptops when my own laptop was broken.

Ruan Müller and Herman Fidder for being true friends that gave any support one can think of.

The families Landman and Müller for their support with appliances and furniture.

My parents and my girlfriend Bianca for their support throughout the whole duration of my studies.

My stay in South Africa was supported by the Landesstiftung Baden-Württemberg with the Baden-Württemberg-STIPENDIUM.

Abstract

Thermal spraying is a technology, which is used for coating of components and structures in order to achieve certain tribological characteristics, or for protection against corrosion, excessive temperature and wear. Within thermal spray, there are processes, which utilise combustion of liquid fuel to obtain high velocities flows providing, therefore, good adhesion of coating materials to substrates. These include High Velocity Oxygen Flame (HVOF) and High Velocity Air Flame (HVAF) process, of which the former one is widely used as it has been developed for at least two decades, while HVAF is less common. However, some studies indicate that HVAF has a number of advantages over HVOF, including the economic benefits. The thermal spray gun, based on the HVAF process, has been developed before, but the system was controlled manually. Therefore, there is a need to develop a fully automated controller of an HVAF thermal spray system.

Process control of thermal spraying is highly complex as it involves simultaneous control of a number of processes, including; ignition process, combustion process, spraying material melting, as well as control and monitoring of auxiliary equipment.

This paper presents the development of a control system for an HVAF thermal spray system, based on a Microchip PIC microcontroller. The designed control system was applied for controlling of thermal spraying of carbides powders, and provided a reliable ignition and stable combustion process, powder feeding and all other functions of control.

Table of Contents

List of Symbols

A	area
A_{throat}	cross sectional area of the nozzle throat
c_p	specific heat at constant pressure
$c_{p,air}$	specific heat at constant pressure of air
$c_{p,exhasut}$	specific heat at constant pressure of exhaust gas
c_v	specific heat at constant volume
E	total energy
e	specific intrinsic energy
err	error
f_{accmin}	typical minimum sampling frequency for active combustion control
f_i	external specific body force
f_{opcmax}	typical maximum sampling frequency for operating point control
f_{opcmin}	typical minimum sampling frequency for operating point control
$j_{\propto i}$	diffusion mass flow of component $\propto$ in x_i-direction
LHV	lower heating value
$\dot{m}$	mass flow rate
$\dot{m}_{air}$	air mass flow rate
$\dot{m}_{fuel}$	fuel mass flow rate
$\dot{m}_{exhaust}$	exhaust gas mass flow rate
M_{CO2}	molar mass of CO_2
N_c	number of components
n_{CO2}	molar fraction of CO_2

p	pressure
p_a	atmospheric pressure
p_{air}	air pressure
p_{jet}	pressure of the jet at the outlet of the nozzle
p_{crit}	critical pressure
p_{fuel}	fuel pressure
p_{oscc}	controlled pressure amplitude
p_{oscuc}	uncontrolled pressure amplitude
q_i	heat flow in x_i-direction
R	ideal gas constant
R_s	specific gas constant
$R_{s,air}$	specific gas constant of air
S_r	radiation source term
$S_\propto$	chemical production term
t	time
T	temperature
T_{air}	temperature of air
$T_{exhaust}$	temperature of exhaust gas
u	velocity
u_i	velocity component in x_i-direction
$\dot{V}$	volume flow
$\dot{V}_{air}$	air volume flow
$\dot{V}_{fuel}$	fuel volume flow

Y_α	mass fraction of the component α
Y_d	energy loss by dissipation
Ξ_{CO2}	mass fraction of CO_2
α	flow coefficient
κ	ratio of specific heats
ρ	density
ρ_{fuel}	fuel density
τ_{ij}	stress tensor
υ	specific volume

Abbreviations

ACC	Active Combustion Control
ADC	Analog Digital Converter
AI	Artificial Intelligence
AIC	Active Instability Control
ANN	Artificial Neural Network
CFD	Computational Fluid Dynamics
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
GUI	Graphical User Interface
HVAF	High Velocity Air-Fuel
HVOF	High Velocity Oxygen-Fuel
IDE	Integrated Development Environment
IM	Internal Model
LD	Light Detector
LMS	Least Mean Square
LS	Light Source
MDS	Multidimensional Simulation
MOSFET	Metal Oxide Semiconductor Field-Effect Transistor
OMG	Object Management Group
OPC	Operating Point Control
PC	Personal Computer
PLC	Programmable Logic Controller

RAM	Random Access Memory
SEM	Scanning Electron Microscope
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SysML	OMG Systems Modeling Language
UART	Universal Asynchronous Receiver Transmitter
UML	Unified Modeling Language

List of Figures

1 Introduction

To get familiar with the topic, a short introduction on combustion control and thermal spraying, particularly the HVAF thermal spray process is given in section 1.1 “Combustion and Thermal Spraying”. Afterwards the purpose of the project is explained in section 1.2 “Purpose of the Project”. The final section of this chapter 1.3 “Organisation of the Dissertation” is an overview on the structure of this dissertation.

1.1 Combustion and Thermal Spraying

Combustion is one of the oldest technologies of mankind and it is still the most important method of energy conversion. (1) While in automotive applications feedback control of combustion is already standard, most of today's combustion systems are controlled without feedback in an open loop mode. The combustion parameters are usually set by determining the required power output and using a lookup table. A precise optimization of the process is not possible with an open loop control. In advanced systems the absence of closed loop control poses serious problems. New combustion technologies will have to integrate feedback controllers, which will allow a fine tuning of the operating conditions. (2)

Thermal spraying is a 100 year old technology, which is used for coating of components and structures in order to achieve certain tribological characteristics, or for protection against corrosion, excessive temperature and wear. Within thermal spray, there are processes, which utilise combustion of liquid or gaseous fuel to obtain high velocities flows providing, therefore, good adhesion of coating materials to substrates. These include High Velocity Oxygen Flame (HVOF) and High Velocity Air Flame (HVAF) process, of which the former one is widely used as it has been developed for at least two decades, while HVAF is less common. (3), (4)

The High Velocity Air Flame Thermal Spraying Process uses a thermal-gun to heat up the spraying material and to obtain high flow velocities. This thermal-gun is also referred to as HVAF-gun and is essentially a ram-jet running on liquid fuel and air.

The schematic diagram of a HVAF thermal spraying gun is shown in Figure 1.1.

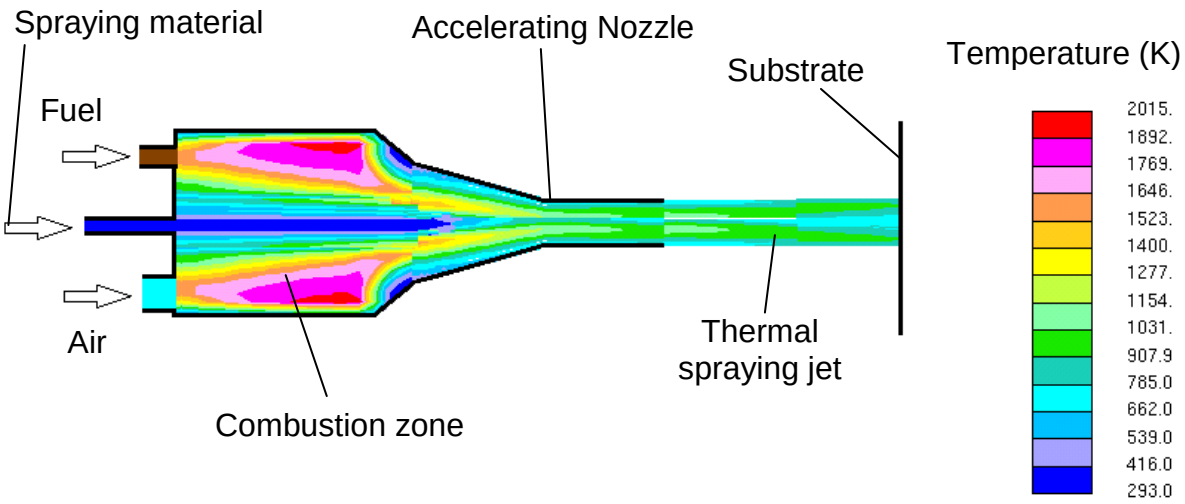


Figure 1.1: Diagram of an HVAF thermal spraying process (?)

In HVAF, similar to gas turbines, a combustion process can be realised in a wide temperature range, depending on the fuel/air ratio. If the fuel/air ratio is lower than the stoichiometric value*, the process is referred to as lean combustion, and it is typically used in the HVAF spray process in order to prevent overheating of the combustor and the accelerating nozzle. The operation of the HVAF-gun is based on an air-standard Brayton cycle.

That is a compression stage since compressed air is supplied to the HVAF-gun by an external source e.g. a diesel or electric compressor, combustion of fuel and air at constant pressure in the combustion chamber, expansion of the exhaust gases in an accelerating nozzle and heat rejection outside of the thermal-gun. (Figure 1.3) (5)

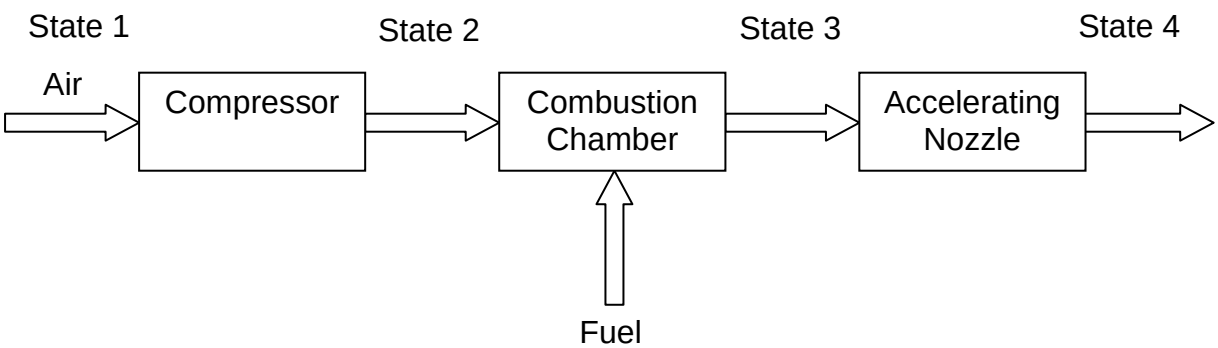


Figure 1.2: Elements and Stages of a thermal-gun system (5)

* The stoichiometric value of a fuel is the specific fuel/air ratio, where a complete combustion occurs, i.e. there are no reactants left over. (1)

To start the combustion an automotive flame glow plug is used. When the combustion has stabilised, the flame glow plug and the fuel supply to the flame glow plug can be switched off, as the combustion is then self-sustained.

As stated before HVAF is less common than HVOF, but HVAF has a number of advantages over HVOF. The use of pure oxygen as oxidant for the combustion in HVOF systems and the setting of the fuel/oxygen ratio to the theoretical stoichiometric value can cause flame temperatures up to T_f = 3000 K. Hence HVOF-guns are water cooled. While HVAF systems run on compressed air on a fuel/air ratio that is lower than the stoichiometric value, they can be realised with air cooled thermal-guns. These air cooled systems have significantly cheaper running costs, as only a compressor is required instead of a water cooling system and pure oxygen. (6) Furthermore the lower process temperature in HVAF thermal spray systems leads to better coating quality. (7)

1.2 Purpose of the Project

The main factor, ultimately determining the quality of coatings in the HVAF thermal spraying process, is the velocity of the thermal jet, since the higher the kinetic energy of the thermal jet and therefore of the particles, the higher the bond strength, coating density and microhardness. This jet velocity depends on the accelerating nozzle geometry and the combustion process parameters. In addition, a high enthalpy jet provides partial or full melting of spraying materials (powders or wires), which temperature also depends on the combustion process parameters. Therefore, there is a need to precisely control the combustion process in the HVAF-gun to achieve constant quality coatings.

By controlling the combustion process parameters manually as in previous systems (6), only a rough control of the combustion process is possible, e.g. fluctuations of the pressure of compressed air can't be taken into account in real time. So the development of a fully automated controller for the HVAF thermal spray system is necessary to achieve constant quality coatings.

The aim of this project is to develop a control system for the HVAF thermal spray process that is capable of providing a stable control of combustion within the HVAF-gun. This includes an automated ignition, the display of the relevant process variables and the control of auxiliary machinery like powder or wire feeders.

Furthermore it will be a base for further research on the HVAF thermal spray process. Therefore the control system shall be expendable for additional equipment, e.g. sensors. The operator shall be able to manually set the operating point of the combustion process and the controller shall be able to capture and store the relevant data on the process. To analyse the captured process data on an external computer an appropriate interface to this computer shall be implemented. This includes the development of software for the external Computer that allows the user to download the process data from the control system for the HVAF thermal spray process and to save this process data on any storage device of the external computer. The developed control system for the HVAF thermal spray process will be a prototype on which different control concepts for the HVAF thermal spray process can be tested. It will operate in lab and industrial environments.

1.3 Organisation of the Dissertation

To get deeper into the topics of combustion control and the control of thermal spraying processes, namely HVOF, a detailed literature search is performed in chapter 2 “Literature Search”. This gives the reader insight into common methods and the state of the art.

In chapter 3 “Theoretical Modelling of Combustion in the HVAF gun”, the combustion process inside the HVAF gun is modelled theoretically. The results of this chapter are the base for the decisions in chapter 4 Control System Design.

The design of the control system for the HVAF thermal spray process is described in chapter 4 “Control System Design”. This includes the design of the mechanical system, the electrical system and the software framework. The software framework acts as base for the execution of control algorithms.

In chapter 5 ”Modelling of the HVAF Thermal Spraying System” the HVAF thermal spraying system is modelled using Matlab/Simulink and the derived model is verified using measurement data obtained with the system described in chapter 4.

Finally in chapter 6 “Conclusion” a conclusion on the results of this study is drawn.

2 Literature Search

To give an overview on the state of the art of combustion modelling and control a detailed literature search is performed.

Common methods for the modelling and simulation of combustion control systems are shown in section 2.1 ”Modelling and Simulation of Combustion Control Systems”.

Control concepts for the control of combustion are described in section 2.2 “Combustion Control”.

The use of artificial intelligence for the modelling and control of combustion is described in section 2.3 “Artificial Intelligence for Combustion Modelling and Control”.

An overview on possible sensor technologies for the control of combustion is given in section 2.4 “Sensors for Combustion Control”.

In section 2.5 “Modelling and Control of High Velocity Oxygen Flame Thermal Spray Processes” a brief look on recent research on modelling and control of HVOF thermal spray processes is taken. For HVAF thermal spray processes no publications related to research on controlling the process could be found.

2.1 Modelling and Simulation of Combustion Control Systems

In the analyses of generic situations of combustion processes, many recent efforts have tried to use control concepts. A detailed modelling would involve the solution of complex Navier-Stokes equations which requires large computing power. Because of this, mainly Low-order modelling is performed. To formalize the control problem and allow theoretical control analysis, block diagrams are used. (2)

A few examples are gathered in Figure 2.1. In Figure 2.1 (a) and (b) the combustion process and the acoustical coupling are distinguished, implying that the path between actuator and sensor is also the path which induces the coupled motion in the unstable operation of the system, though this is not always the case. In many cases the actuator modulates a secondary fuel injection and the chemical conversion of the injected fuel induces an acoustic wave. This wave then combines with the acoustic motion associated with the instability in the system. A more detailed model that takes these circumstances into account is shown in Figure 2.1 (d); LMS is a least

mean square filter. For a model without a detailed description of the plant using an adaptive controller, see Figure 2.1 (c). (2)

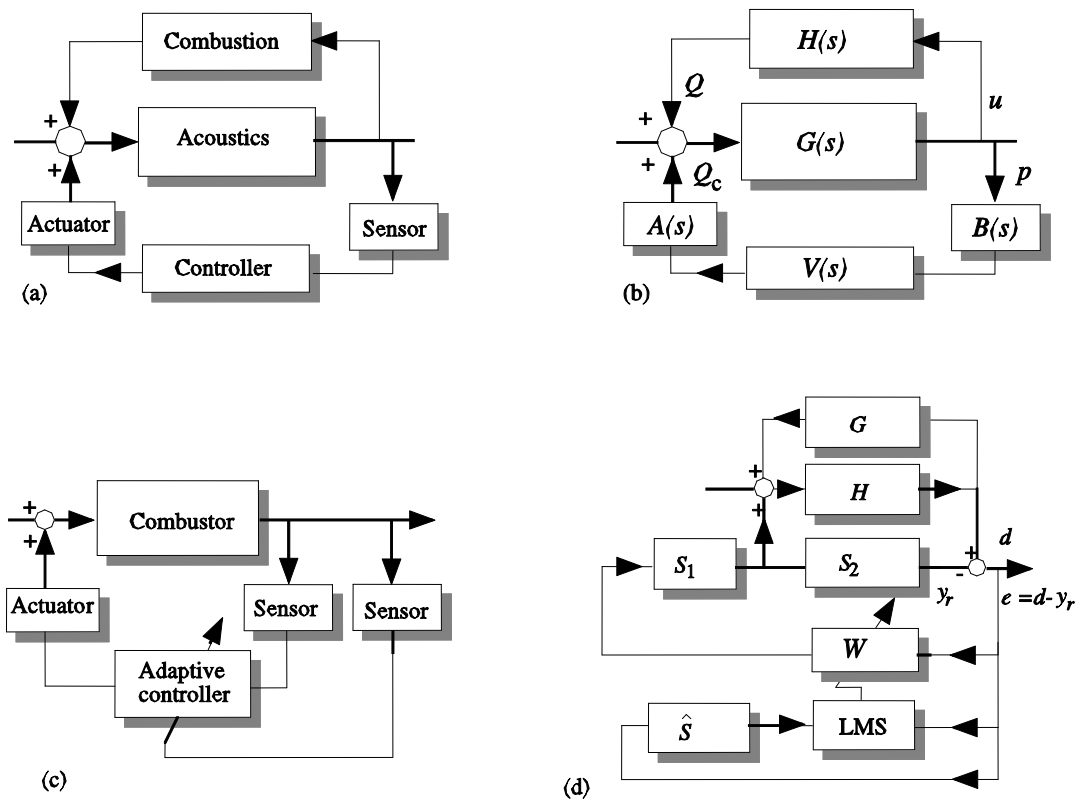


Figure 2.1: Control block diagrams (2)

Many studies also try to look at the problem from the point of view of the control theory. For example state space descriptions of combustor and controller in a robust control framework are used (2).

A relatively new approach in simulation of combustion control systems is multidimensional simulation (MDS). For MDS a unsteady Navier-Stokes flow solver is coupled with a control algorithm. This allows complete software studies of control concepts. A MDS is discussed by Candel (2) using the control of vortex-driven instabilities found in solid rocket motors as an example. Figure 2.2 (a) shows the principle and Figure 2.2 (b) shows the coupling between the flow solver Sierra and the adaptive controller “ACR”.

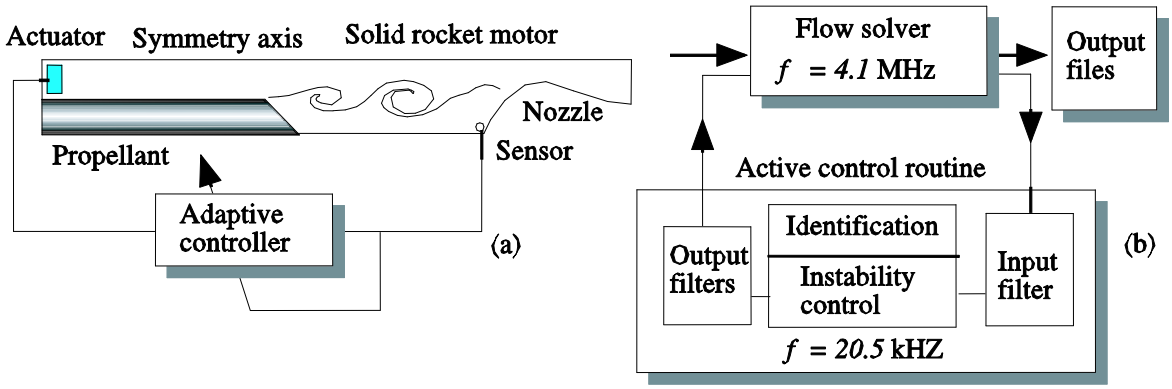


Figure 2.2: Active control of a solid rocket motor (2)

By coupling a flow solver with a control algorithm, some problems arise. The two most important are, first the representation of the actuator which would be used to drive the flow and second the mismatch between the time stepping of the flow solver and the sampling period of the controller. Therefore MDS of a combustion control system requires a suitable description of the actuator and a careful coupling of the flow solver and the controller. One solution for the first problem is to represent the actuator with a distribution of sources. To deal with the coupling problem of the flow solver and the controller, the input and output of the controller has to be suitably filtered. (2)

2.2 Combustion Control

Concepts for combustion control can be divided in two categories (8):

- Operating Point Control (OPC)

  To maintain certain flame parameters like the equivalence ration of fuel to air in a prescribed range of values, the injection of fuel is regulated.

  Typical frequency range: $f_{opcmin} = 1\ Hz$ to $f_{opcmax} = 100\ Hz$

- Active Combustion Control (ACC)

  To improve the combustion characteristics or to avoid/limit pressure oscillations, the flow properties (e.g. fuel flow rate) are modulated by the controller.

  Typical frequency range: $f_{accmin} = 20\ Hz$ to a few kHz

2.2.1 Operating Point Control

As mentioned in chapter 1 Introduction, feedback control is standard in automotive internal combustion engines.

Figure 2.3 shows a typical operating point control system for an automotive engine.

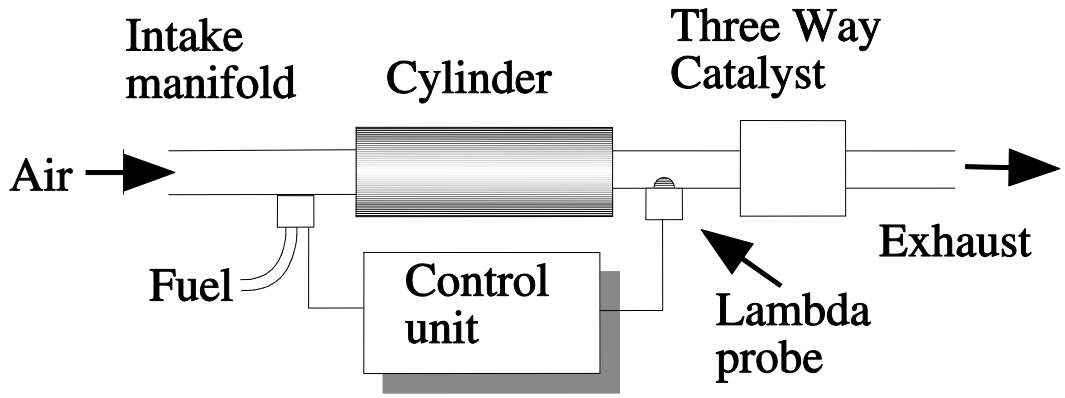


Figure 2.3: Conventional automotive engine control system (8)

To meet exhaust-pollution standards catalytic converters are used, that oxidise excess levels of exhaust carbon monoxide and hydrocarbons simultaneously and reduce excess levels of nitrogen oxides. Because of its effect on all three pollutants, these devices are usually referred to as three way catalysts. Standard three way catalysts have the best conversion efficiency close to stoichiometric conditions, so the control system has to maintain stoichiometric combustion to ensure efficient catalytic conversion and clean engine operation. To achieve this, a “lambda” probe is used (designates the air to fuel equivalence ratio). It is placed in the exhaust stream and measures the oxygen content of the exhaust gas. As the air to fuel ratio is uniquely related to the oxygen level in the exhaust gas, the control unit can adjust it to stoichiometric conditions. (9)

An operating point control of a premixed gas turbine combustor is shown in Figure 2.4. The operating point is adjusted through fuel regulation. The fuel demand is calculated using air flow, hygrometry and fuel properties measurements. (2)

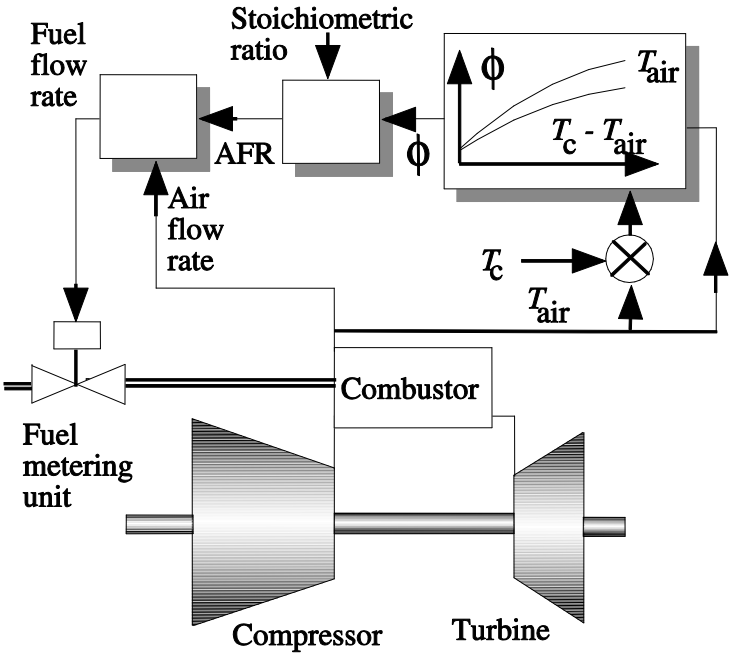


Figure 2.4: Operating point control of a premixed gas turbine combustor (2)

2.2.2 Active Combustion Control

An active instability suppression using periodic liquid-fuel injection is discussed by Yu, Wilson and Schadow (10). The experimental set-up is shown in Figure 2.5. A controller fuel is pulsed directly into the combustion chamber, while the injection timing is adjusted with respect to the combustor pressure signal. The controller fuel makes up 12%-30% of the total heat release.

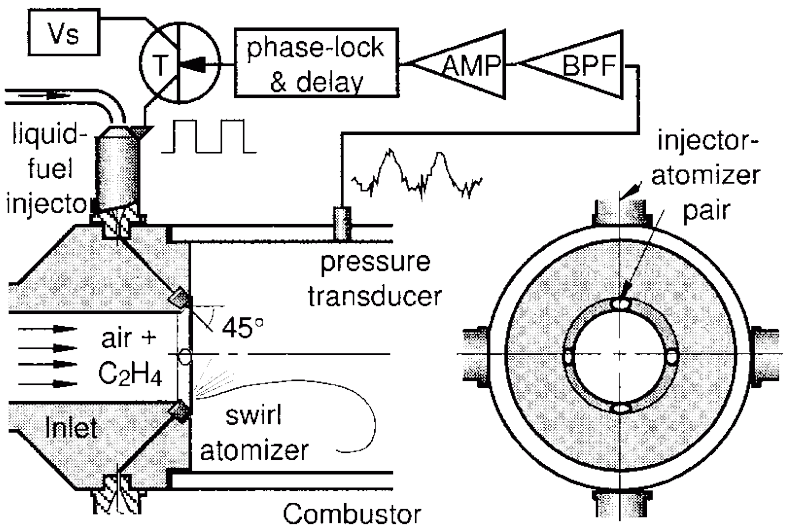


Figure 2.5: Active combustion control system (10)

This set-up of simple closed-loop control was applied to two different cases that developed natural instabilities. In the first case, a 70 kW combustor, at uncontrolled pressure amplitude of p_{oscuc} = 356 Pa, the closed-loop control allowed up to p_{oscc} = 292 Pa reduction of the pressure amplitude, while the instability frequency was not affected by the controller. In the second case, a 270 kW combustor, the injection timing affected both instability frequency and amplitude. For the second case, the closed-loop controller was not able to maintain the oscillation amplitude on a suppressed level, resulting in unsteady modulation of the oscillation amplitude and frequency. The intermittent loss of control was linked to the frequency-dependent phase shift, associated with an electronic band-pass filter. The results show the limitation of a simple phase-delay approach in completely suppressing the natural oscillations under certain conditions. (10)

Similar to Yu, Wilson and Schadow (10) an active instability control is presented by Hermann et al. (11), but for a much larger scale application. An ACC is implemented on a 260 MW heavy duty gas turbine.

2.3 Artificial Intelligence for Combustion Modelling and Control

The traditional modelling and control of combustion involves the solution of complex differential equations. For solving these differential equations complicated algorithms that usually require large computing power and time are used. While Artificial intelligence (AI) systems don't require complex rules and mathematical routines, AI systems are able to learn the key information patterns within a multi-dimensional information domain. Additionally these systems have characteristics that are beneficial for a use with combustion processes. Artificial Neural Networks (ANN) for example are fault tolerant, robust and noise immune. (12)

Kalogirou (12) gives an overview of 59 applications where AI systems have been used for modelling and/or control of combustion systems and internal combustion engines.

An active control of combustion instabilities on a Rijke tube is presented by Blonbou et al. (13). This Rijke tube presents for some operating conditions instabilities with pressure amplitude of up to p_{oscuc} = 356 Pa. An internal model control scheme for nonlinear systems that uses two ANNs has been developed to control these instabilities. The internal model (IM) is realised with the first ANN and approximates

the system forward dynamics. The controller is realised with the second ANN and calculates the control input. The parameters of the controller are updated adaptively in real time. The IM was first trained to reproduce the burner response (given by the pressure or heat-release measurements) to open loop excitation. After the IM has been trained, it was used in the control loop to predict the response of the burner to the control action. This prediction was used by the adaptive control algorithm to update the parameters of the controller. The developed controller is able to attenuate the instabilities in real time for fixed or variable operation conditions and damping of the pressure amplitude down to p_{oscc} = 3,6 Pa has been obtained. (13)

2.4 Sensors for Combustion Control

There is a range of sensors that can be used for combustion control, including but not limited to:

- Pressure sensors,
- Temperature sensors,
- Gas sensors,
- Flow sensors,
- Optical sensors.

A generic combustion process with available or promising sensing techniques is shown in Figure 2.6.

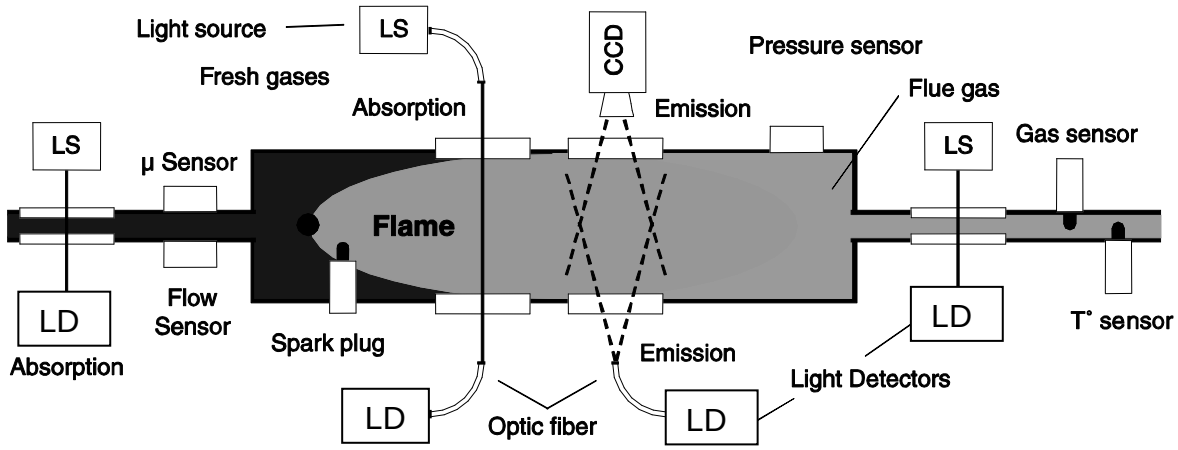


Figure 2.6: Typical combustion process, sensors and diagnostic techniques for combustion control (8)

Selecting a sensing technique depends mainly on the practical aspect of applying the sensors to the combustion process. For example optical sensors require at least one optical access, which is often difficult to incorporate in a practical device.

For sensors that are directly attached to the combustor it is of prime importance that they must be able to withstand the hostile conditions prevailing in the combustor. (8)

2.5 Modelling and Control of High Velocity Oxygen Flame Thermal Spray Processes

Current research on the modelling and control of HVOF thermal spray processes ((14), (15)) goes one step further than classic combustion modelling and control, as it takes the multiscale character of an HVOF thermal spray process into account, see Figure 2.7. The objective of this research is to precisely control the coating micro- and nano-structure that determines the coating mechanical and physical properties, by developing computational methodologies that manipulate the macroscale operating conditions such as the gas flow rate and spray distance.

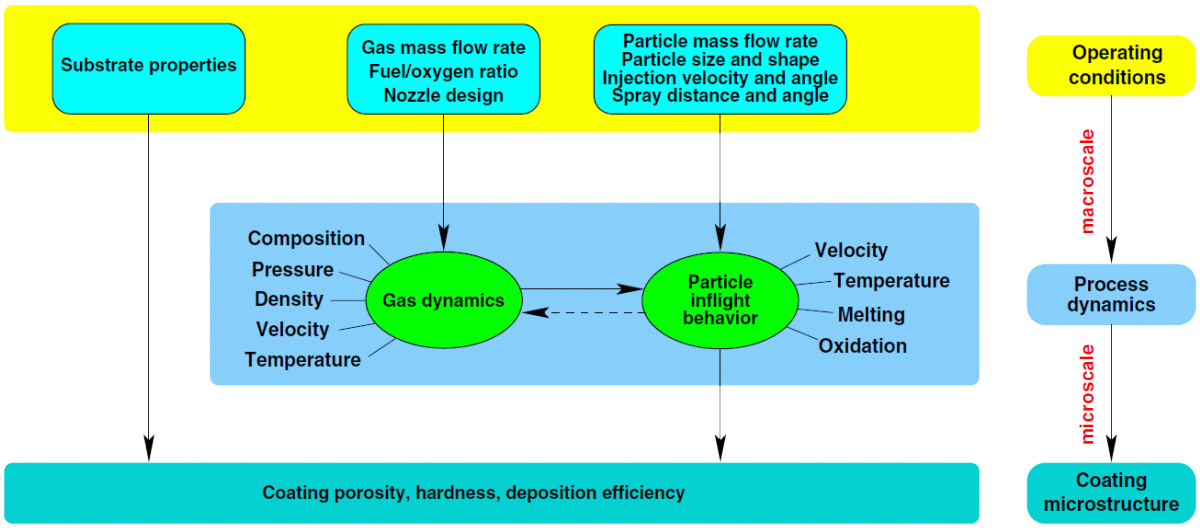


Figure 2.7: Multiscale character of the HVOF thermal spraying process (14)

Li, Shi and Christofides (15) present a fundamental model and a feedback control system for an industrial HVOF thermal spray process (Diamond Jet hybrid gun, Sulzer Metco, Westbury, NY). The model was validated with data from existing experimental studies and was set as basis for the formulation of the control problem

for this HVOF process. The feedback controller was tested on the process model and it performed well. The proposed control system for the real process, shown in Figure 2.8, incorporates an optical measuring device, that provides online measurement of individual particle characteristics including temperature, velocity and size (14). According to these measurements the feedback controller adjusts the flow rates of fuel, oxygen and air to achieve the desired set-point values (15).

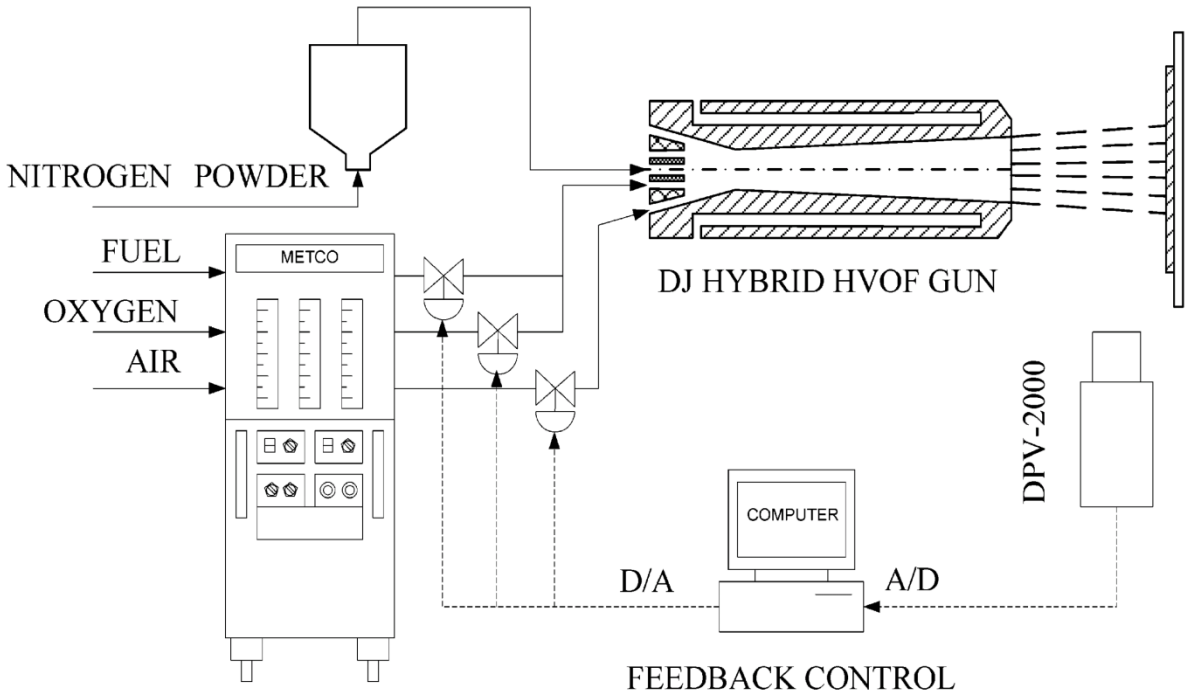


Figure 2.8: Schematics of the proposed HVOF feedback control system (15)

3 Theoretical Modelling of Combustion in the HVAF Gun

For the design of the control system, especially for the selection of actuators and sensors, it is important to understand the physical principles behind the process. If an inappropriate set of sensors and actuators is chosen it might be difficult or impossible to control the process. Therefore the combustion inside the HVAF gun and the effect of the accelerating nozzle are modelled theoretically.

The governing equations for a combustion process are shown in section 3.1 “Governing Equations”.

A low order approach for modelling combustion is described in section 3.2 “Low Order Modelling of Combustion”.

The working principle of the accelerating nozzle and its influence on the combustion inside the combustion chamber are explained in section 3.3 “The Accelerating Nozzle”.

3.1 Governing Equations

The combustion within the HVAF gun is a reactive flow which can be described with the following equations under the conditions that:

- The fluid is a continuum,
- Thermal equilibrium is present,
- The fluid properties are isotropic,
- The fluid is a Newtonian fluid,
- Stokes’ law, Fourier’s law and Fick’s law are valid,
- The fluid is an ideal gas.

Using tensor notation and conservative formulation, a system of coupled partial differential equations is generated (16):

Total mass:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}\,(\rho u_i) \;= 0 \tag{3.1}$$

Component masses:

$$\frac{\partial}{\partial t}(\rho Y_{\propto}) + \frac{\partial}{\partial x_i}(\rho u_i Y_{\propto}) + \frac{\partial j_{\propto j}}{\partial x_i} = S_{\propto} \tag{3.2}$$

Momentum:

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) - \frac{\partial \tau_{ij}}{\partial x_j} + \frac{\partial p}{\partial x_i} = \rho f_i \tag{3.3}$$

Energy:

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_i}(\rho u_i E + u_i p) - \frac{\partial}{\partial x_i}(u_i \tau_{ji}) + \frac{\partial q_i}{\partial x_i} = \rho u_i f_i + S_r \tag{3.4}$$

With $i, j = 1, 2, 3$ and $\propto = 1, 2, \dots, N_{k-1}$

The Einstein summation convention applies exclusively to the Latin indices from i to m.

u_i is the component in x_i-direction of the velocity vector:

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \tag{3.5}$$

p is the pressure, Y_α is the mass fraction of the component α in the mass fraction vector $\mathbf{Y} = \left(Y_1, Y_2, \dots, Y_{N_c}\right)^T$, N_c is the number of the components.

The specific total energy is:

$$E = e + \frac{\mathbf{u}^2}{2} \tag{3.6}$$

Where e is the specific intrinsic energy. The enthalpy of formation of the gas fractions is included in e. Because of that there is no reaction source term in equation (3.4).

To solve this system of equations the heat flow q_i in x_i-direction, the diffusion mass flow $j_{\alpha i}$ of the component α in x_i-direction, the external specific body forces f_i , the radiation source term S_r , the elements of the stress tensor τ_{ij} and the chemical production term S_α of the component α need to be calculated from the dependent variables.

3.2 Low Order Modelling of Combustion

As one can see from the previous section, the governing equations for a combustion process are not very handy for a quick identification of the main parameters influencing the process. Therefore a low order approach for the modelling of combustion is sufficient.

In section 3.2.1 “Adiabatic Flame Temperature” the concept of such a low order approach is explained.

The composition of the exhaust gas is a factor influencing the operation of the HVAF gun and is described in section 3.2.2 “Composition of the Exhaust Gas”.

3.2.1 Adiabatic Flame Temperature

The adiabatic flame temperature is the temperature that would be achieved if the combustion occurred in an adiabatic, hence in an ideal insulated combustion chamber. Because no heat exchange occurs with the environment, the temperature of the exhaust gas is the same as the flame temperature. In a real application, there is of course always a heat exchange with the environment. Hence the temperatures of the exhaust gas calculated with this method will be higher than in reality.

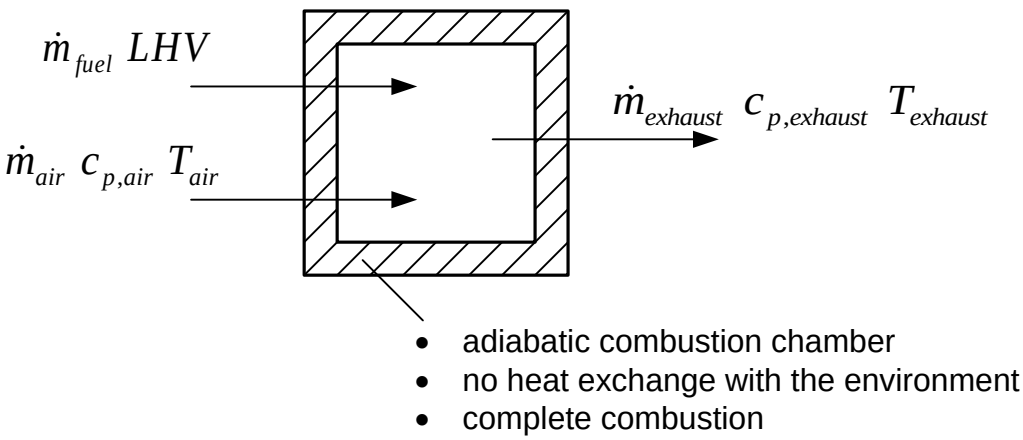


Figure 3.1: Energy and mass balance for the calculation of the adiabatic flame temperature (17)

With the assumption that the combustion is complete, thus the fuel is completely oxidised, the energy balance is (17):

$$\dot{m}_{fuel}\, LHV \; + \; \dot{m}_{air}\, c_{p,air}\, T_{air} \; = \; \dot{m}_{exhaust}\, c_{p,exhaust}\, T_{exhaust} \tag{3.7}$$

Where $\dot{m}_{exhaust}$ is the mass flow rate of the exhaust gas. $c_{p,air}$ and $c_{p,exhaust}$ are the specific heats of air and the exhaust gas. T_{air} and $T_{exhaust}$ are the temperatures of air and the exhaust gas. LHV is the lower heating value of the fuel.

With the conservation of mass:

$$\dot{m}_{exhaust} = \dot{m}_{fuel} + \dot{m}_{air} \tag{3.8}$$

Where $\dot{m}_{fuel}$ and $\dot{m}_{air}$ are the mass flow rates of fuel and air.

Equation (3.7) can be changed that the temperature of the exhaust gas can be calculated:

$$T_{exhaust} = \frac{LHV + \frac{\dot{m}_{air}}{\dot{m}_{fuel}}\, c_{p,air}\, T_{air}}{c_{p,exhaust}\left(1 + \frac{\dot{m}_{air}}{\dot{m}_{fuel}}\right)} \tag{3.9}$$

From equation (3.9) it can be seen, that the main parameter for influencing the operating point of a combustion process is the ratio of fuel and air. As the other parameters either depend on the ratio of fuel and air or are given by the conditions of fuel and air.

3.2.2 Composition of the Exhaust Gas

The composition of the exhaust gas is influencing the adiabatic flame temperature and it also affects the operating conditions of the accelerating nozzle described in section 3.3. Therefore a closer look is taken on the properties of the exhaust gas and how they can be calculated.

It is assumed that the reaction between fuel and air can be described by a single global reaction. In the case of kerosene this is:

$$C_{12}H_{24} \; + \; 24\, O_2 \; \rightarrow \; 12\, CO_2 + \; 12\, H_2O \tag{3.10}$$

In reality the combustion of kerosene and air consist of several elementary reactions and would also result in the generation of NO, NO_2 and CO. (18)

The specific heat of the exhaust gas can be calculated using the mass fractions of the exhaust gas components (19):

$$c_{p,exhaust} = \Xi_{N2}\, c_{p,N2} + \Xi_{O2}\, c_{p,O2} + \Xi_{CO2}\, c_{p,CO2} + \Xi_{H2O}\, c_{p,H2O} \tag{3.11}$$

Where Ξ_{N2}, Ξ_{O2}, Ξ_{CO2}, Ξ_{H2O} are the mass fractions and $c_{p,N2}$, $c_{p,O2}$, $c_{p,CO2}$, $c_{p,H2O}$ are the specific heats of N_2, O_2, CO_2, H_2O.

Similar to the specific heat calculation, the molar mass of the exhaust gas can be calculated by using the molar fractions of the exhaust gas components (17):

$$M_{exhaust} = n_{N2}\, M_{N2} + n_{O2}\, M_{O2} + n_{CO2}\, M_{CO2} + n_{H2O}\, M_{H2O} \tag{3.12}$$

Where n_{N2}, n_{O2}, n_{CO2}, n_{H2O} are the molar fractions and M_{N2}, M_{O2}, M_{CO2}, M_{H2O} are the molar masses of N_2, O_2, CO_2, H_2O.

3.3 The Accelerating Nozzle

As mentioned in chapter 1 the main elements of an HVAF gun are the combustion chamber and the accelerating nozzle. To determine the influence of the conditions outside the HVAF gun on the combustion inside the combustion chamber, the basic working principles of the accelerating nozzle are explained. Looking at an idealised one-dimensional description of the nozzle is sufficient for understanding these basic working principles.

Idealised in this context is:

- One-dimensional form of the continuity.
- The velocity at the inlet of the nozzle is negligible small compared to the velocity at the outlet of the nozzle.
- The expansion in the nozzle is isentropic.

A convergent nozzle can expand a flow only to a certain pressure, the so called critical pressure p_{crit}. Hence it can also accelerate a flow only up to a certain speed

and not further. This maximum speed is the local speed of sound of the fluid that is expanded. The critical pressure p_{crit} can be calculated as follows (20):

$$p_{crit} = \left(\frac{2}{\kappa + 1}\right)^{\frac{\kappa}{\kappa-1}} p_1 \tag{3.13}$$

Where p_1 is the pressure at the inlet of the nozzle and κ is the ratio of specific heats of the fluid:

$$\kappa = \frac{c_p}{c_v} \tag{3.14}$$

Where the specific heat at constant pressure is c_p and the specific heat at constant volume is c_v.

If the atmospheric pressure p_a is lower than the critical pressure p_{crit}, the jet at the exit of the nozzle suddenly expands. Due to inertia the jet expands above equilibrium. Now the atmospheric pressure is higher than the pressure in the jet, leading to a sudden compression of the jet. This carries on until the energy resulting from the overpressure is consumed by friction and eddies. The expansion and compression waves in the jet result in so called shock diamonds (Figure 3.1) coupled with high noise emission.

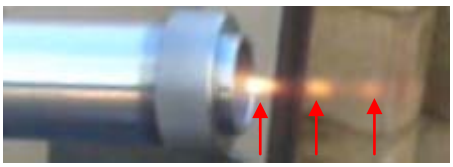


Figure 3.2: HVAF gun without divergent section of the de Laval nozzle, jet with shock diamonds

Because the HVAF thermal spraying process requires supersonic jet velocities, a convergent nozzle cannot be used. Therefore the accelerating nozzle is a de Laval nozzle (Figure 3.2). That is a nozzle which consists of a convergent section (to accelerate the subsonic flow to sonic velocity) and a divergent section (to further accelerate the flow to supersonic). At the nozzle throat, where the cross sectional

area is a minimum, the flow velocity is the sonic velocity (Mach number $Ma = 1$) and the pressure is the critical pressure.

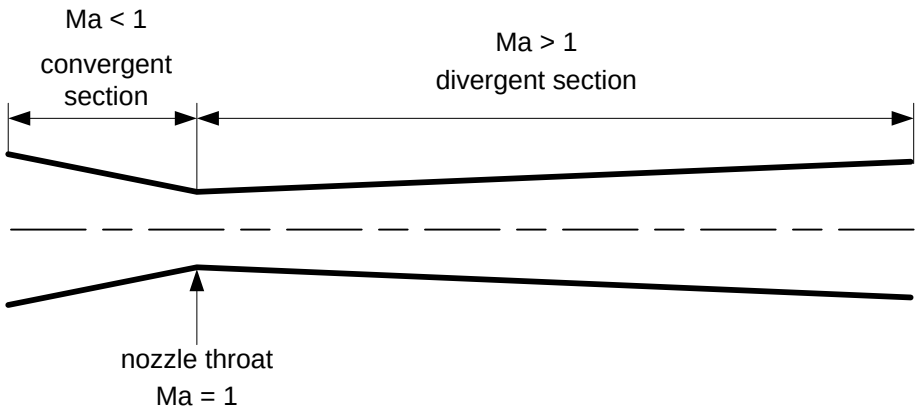


Figure 3.3: Cross section of a de Laval nozzle

The jet velocity at the exit of a de Laval nozzle can be calculated with the following formula (20):

$$u_2 = \left[2\frac{\kappa}{\kappa - 1}RT_1\left\{1 - \left(\frac{p_2}{p_1}\right)^{\frac{\kappa-1}{\kappa}}\right\}\right]^{\frac{1}{2}} \tag{3.15}$$

Where T_1 and p_1 are temperature and pressure at the inlet of the nozzle, κ is the ratio of specific heats of the fluid, R is the ideal gas constant and p_2 is the pressure at the exit of the nozzle, hence the atmospheric pressure.

The flow in the divergent section of the nozzle is supersonic as long as the atmospheric pressure p_a is smaller than the critical pressure p_{crit}. In this case a change of the atmospheric pressure p_a has no influence on the ratios inside the nozzle, as the signal of pressure change cannot travel upstream in a supersonic flow. Hence the combustion in the combustion chamber is not influenced by atmospheric pressure fluctuations as long as the condition $p_a < p_{crit}$ is fulfilled.

For a given nozzle design and certain inlet temperature T_1 and pressure p_1 there is only one pressure value of the atmospheric pressure p_a where the jet is a

supersonic, parallel jet. This is the case when p_a is equal to the pressure of the jet at the outlet of the nozzle p_{jet}.

If the atmospheric pressure p_a is different from the pressure of the jet p_{jet} and smaller than the critical pressure p_{crit} the jet is also supersonic, but due to the difference between p_a and p_{jet} shock diamonds and high noise emission occur. This is the typical operating condition of an HVAF gun and can be seen in Figure 3.3.

Figure 3.4: HVAF gun with de Laval nozzle, supersonic jet with shock diamond

Beside the velocity generated by the accelerating nozzle, it is also important to look at the mass flow rate through the nozzle.

If the flow through a de Laval nozzle is supersonic, thus $p_a < p_{crit}$, the mass flow rate through the nozzle can be calculated by (20):

$$\dot{m}_{nozzle} = A_{throat} \left[2 \frac{p_1}{\upsilon_1} \frac{\kappa}{\kappa + 1} \right]^{\frac{1}{2}} \left[\frac{2}{\kappa + 1} \right]^{\frac{1}{\kappa - 1}} \tag{3.16}$$

Where A_{throat} is the cross-sectional area at nozzle throat, p_1 is pressure at the inlet of the nozzle, κ is the ratio of specific heats of the fluid and υ_1 is the specific volume of the fluid.

One can see that, under the given conditions, the mass flow rate through the nozzle only depends on the smallest cross-sectional area of the nozzle and on the properties of the exhaust gas. There is no influence of the ambient conditions.

4 Control System Design

After the requirements for the control system, the state of the art and the theoretical fundamentals of combustion have been clarified, the design of the actual control system is next.

To design and model the control system and to refine the requirements for the control system the Object Management Group[†] (OMG) Systems Modeling Language 1.1 (SysML) and the Unified Modeling Language 2.2 (UML) are used. Therefore in section 4.1 “UML and SysML” a short introduction on these two related modelling languages is given.

The selection of actuators, sensors and of a controller is described in section 4.2 “Selection of Components”.

In section 4.3 “System Architecture” an overview is given over the whole control system.

This is followed by the detailed design of the control system, starting with the fluid mechanical and the electrical system in section 4.4 “Hardware”.

Finally the developed software framework is discussed in section 4.5 “Software Design and Implementation”.

4.1 UML and SysML

UML is a modelling language that is mainly used for the design of software. Due to the demand for a language that also can be used for modelling systems that not exclusively consist of software, SysML was developed. SysML is an extended subset of the Unified Modeling Language 2.2 (UML), see Figure 4.1.

Due to the fact most of the diagram types used in this project are part of UML and of SysML, a clear separation if the specific model is a UML or SysML model is not possible. It is handled that way, that if a diagram is used to model hardware, it is referred to as SysML model and if a diagram is used to model software it is referred to as UML model. The diagrams used in this project which are the same in UML and

[†] The Object Management Group (OMG) is a computer industry consortium with approximately 800 members such as IBM or Motorala. OMG is amongst other things responsible for the development and standardisation of UML and SysML. (21)

SysML are namely, the block definition diagram, the use case diagram, the activity diagram, the sequence diagram and the state diagram

In Appendix E is a SysML reference which is from Weilkiens (21).

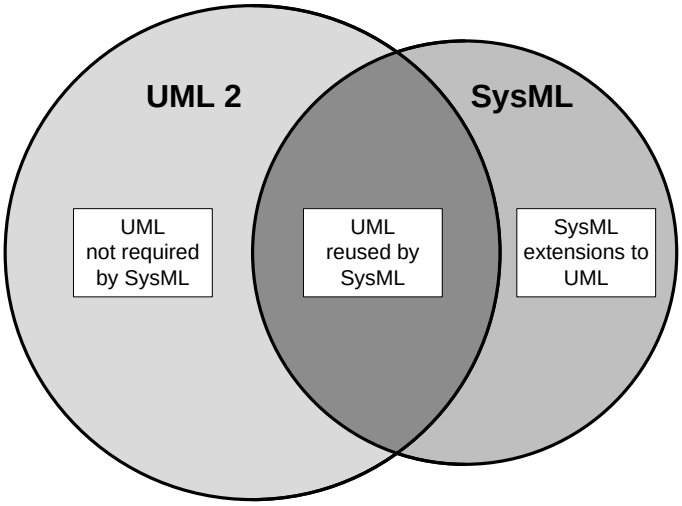


Figure 4.1: UML and SysML (22)

4.2 Selection of Components

4.2.1 Selection of Actuators

It can be seen from the previous chapters that the fuel/air ratio is the parameter that directly influences the combustion. To achieve stable combustion in a certain operating point, the fuel/air ratio needs to be controlled. This can be achieved by regulating the mass flow rates of fuel and/or air by proportional valves. There are three possible options for placing proportional valves in the supply lines of the HVAF gun.

- One proportional valve in the fuel line
- One proportional valve in the air line
- One proportional valve in the fuel line and one in the air line

The criteria for the selection of an installation location are the costs, resulting operating range and the complexity of the control problem formulation. The evaluation of the installation location for proportional valves is shown in Table 4.1

	valve in fuel line	valve in air line	valves in fuel and air lines
Complexity of control	o	o	-
Operating range	o	-	+
Costs	+ +	-	- -

Table 4.1: Evaluation of installation locations for proportional valves

A proportional valve only in the air line is not practical, as the HVAF gun runs in a lean combustion mode. Under certain conditions this can lead to a limitation of the operating range. Also the price for a proportional valve for the air line is quite high, due to the large volume flow in normal operation ($\dot{V}_{air} = 2{,}8\ m^3/min$).

The cheapest and therefore chosen solution is a proportional valve in the fuel line. The complexity of control and the operating range are reasonable for this solution.

The solution that offers the widest operating range is a combination of two proportional valves, one in the fuel line and one in the air line. The disadvantages are the high price for the air valve and increased complexity of control.

Additionally 3/2 valves are built into the fuel and air lines to allow the controller to switch these lines on and off.

4.2.2 Selection of Sensors

As stated in the literature search there are various possibilities of sensing the relevant parameters for combustion control. Any measurement in the exhaust stream is not practical as the sensors are too expensive. Also measurements in the combustion chamber are not practical due to the high temperature (up to 3000K). So there's only the possibility of taking sensor values in the supply lines of the HVAF-gun.

Therefore the pressure and flow rate of air and fuel are measured by sensors. Additionally the air temperature is measured.

This set of sensors allows the calculation of the mass flow rates for air and fuel.

The mass flow rate is defined as follows (20):

$$\dot{m} = \rho A u \tag{4.1}$$

Where ρ is the density, A is the area and u is the velocity.

For air there's also the Ideal gas law applicable (20):

$$\rho = \frac{p}{R_s T} \tag{4.2}$$

Where p is the pressure, T is the temperature and R_s is the specific gas constant

By combining equations (4.1) and (4.2) the air mass flow rate $\dot{m}_{air}$ can be calculated from the air pressure p_{air}, the air temperature T_{air} and the specific gas constant of air $R_{s,air}$.

$$\dot{m}_{air} = \frac{p_{air}}{R_{s,air} T_{air}} \dot{V}_{air} \tag{4.3}$$

The fuel mass flow rate $\dot{m}_{fuel}$ can be directly calculated by equation (4.1) if the change of fuel density ρ_{fuel} over temperature T_{fuel} is assumed to be negligible. $\dot{V}_{fuel}$ is the fuel flow rate.

$$\dot{m}_{fuel} = \rho_{fuel} \dot{V}_{fuel} \tag{4.4}$$

Additionally a current sensor is placed in the power supply line of the flame glow plug, to allow the controller monitoring of the ignition process.

4.2.3 Selection of the Controller

The control system for the HVAF process will be, besides normal operation, used as a base for further research. Therefore the flexibility of the used controller plays an important role. Other criteria for selecting a controller are the price of the system, the time it takes until the controller is shipped and the time and effort it takes for commissioning. The evaluation of different controller types according to these criteria is shown in Table 4.1.

	Microcontroller	PLC	Industrial PC with soft PLC	FPGA
Costs	+ +	-	-	o
Available immediately	+ +	-	-	-
Commissioning	-	+	o	-
Flexibility	+	o	+	+

Table 4.2: Evaluation of different controller types

It can be seen that under these criteria a FPGA based solution would have no advantage over any of the other solutions. Although a PLC or an industrial PC would have advantages in commissioning, a microcontroller is chosen. Mainly because of the fact, that a complete microcontroller board is available immediately, for a reasonable price from the department of Electrical Engineering of the NMMU. This board is the so called multi I/O board and it is equipped with a Microchip PIC24HJ256GP610, 16-bit microcontroller.

4.3 System Architecture

To visualise the architecture and elements of the control system for the HVAF thermal spray process the SysML block diagram shown in Figure 4.2 is used. The components that already existed at the beginning of this project are marked red.

The HVAF control system is a classical mechatronic system consisting of a mechanical system, an electrical system and software.

The mechanical system consists of four subsystems: air system, fuel system, powder system and HVAF gun.

The HVAF gun was developed in previous projects and is, as described in chapter 1.1.1 ”The High Velocity Air Flame (HVAF) Thermal Spraying Process”, responsible for melting and accelerating of the spray material. It consists of a combustion chamber, an accelerating nozzle and a flame glow plug. The flame glow plug is also part of the electrical system, as it generates the heat for igniting the fuel with electric current.

The powder system provides the powder supply to the HVAF gun and consists of a powder feeder, an air dryer and of sensors. The sensors of the powder system are also part of the electrical system.

The fuel system supplies the HVAF gun with fuel for the combustion. It consists of sensors and actuators which are also part of the electrical system.

The air system supplies the HVAF gun with air for the combustion. It consists of sensors and actuators which are also part of the electrical system.

The electrical system consists, besides the already mentioned components, of the interface boards, the user interface, the power supplies and the multi I/O board.

The most important part of the multi I/O board is the microcontroller, on which the microcontroller program is executed. The whole control logic of the control system is implemented in the microcontroller program. Hence the whole control system is controlled by the microcontroller and the program on it.

Due to different operating voltages of the microcontroller and rest of the control panel, the interface boards realise the voltage level shift. The power supplies supply the components of the electrical system with power and the user interface allows operating of the control system.

The software consists of the already mentioned the microcontroller program and of a PC program. The PC program allows the operator to transfer captured process data from the microcontroller to an external PC for further analysis.

A more hardware orientated and less abstract view on the system is shown in Figure 4.3. The already existing components are marked with a green frame in this figure. The details of this illustration are explained in the following section 4.4 “Hardware”.

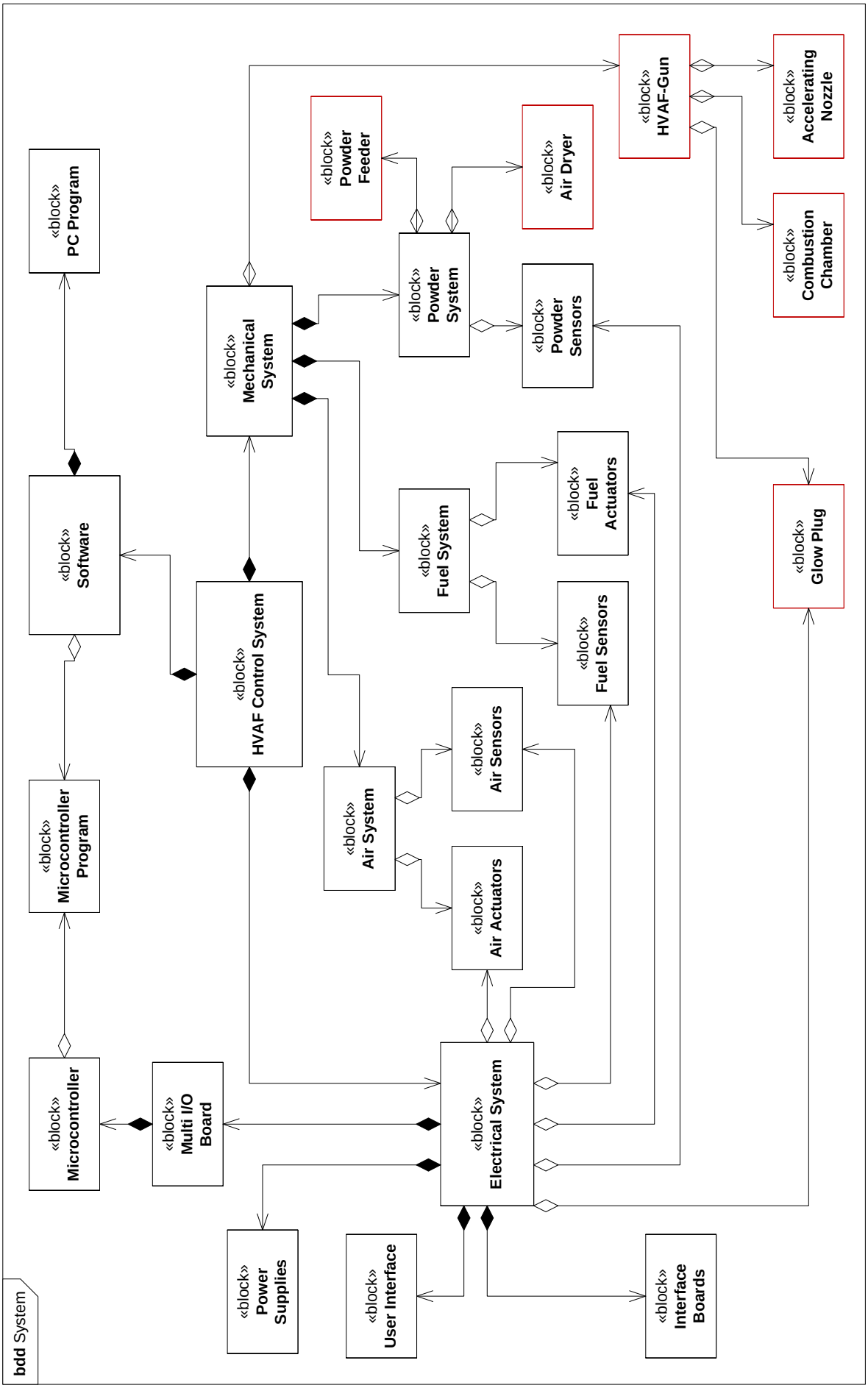


Figure 4.2: Block diagram of the control system for the HVAF thermal spray process

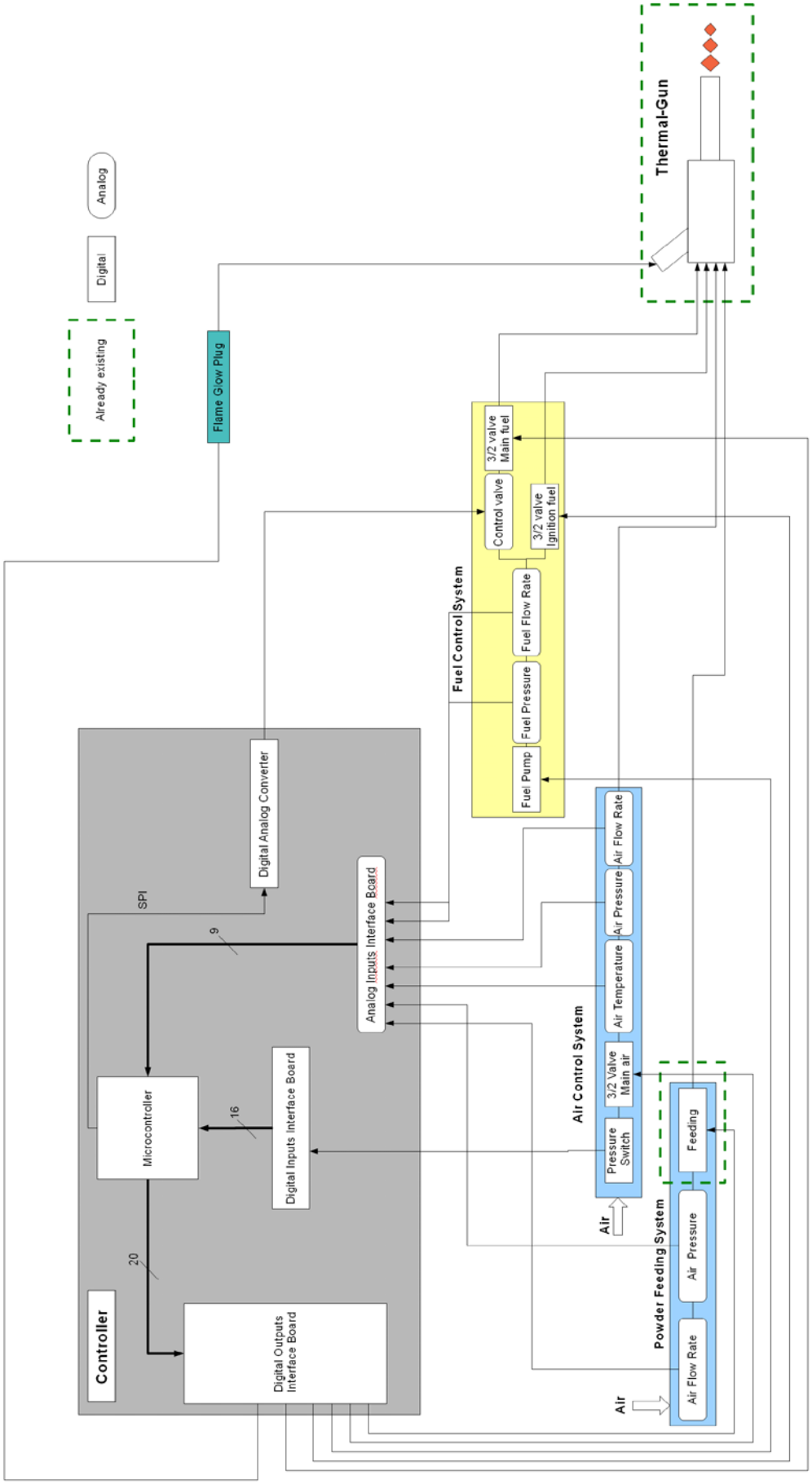


Figure 4.3: Schematic overview of the control system for HVAF thermal spray process

4.4 Hardware

The hardware of the control system can be divided into two subsystems. A mechanical system, which is described in section 4.4.1 “Air and Fuel System” and an electrical system, which is described in section 4.4.2 “Electrical System”.

4.4.1 Air and Fuel System

The air and fuel system consists of three independent subsystems, namely the main air system, the fuel system and the air system for the powder feeder.

The main air system is shown in Figure 4.4. After the main air ball valve and the pressure switch the system is split into three lines. The main line is equipped with a 3/2 valve, an air pressure sensor, an air temperature sensor and an air flow meter. The second line is used to supply the HVAF gun with a small amount of air during ignition. It bypasses the 3/2 valve of the main line with a ball valve. The third line is for special purposes such as substrate cooling and is equipped with a ball valve.

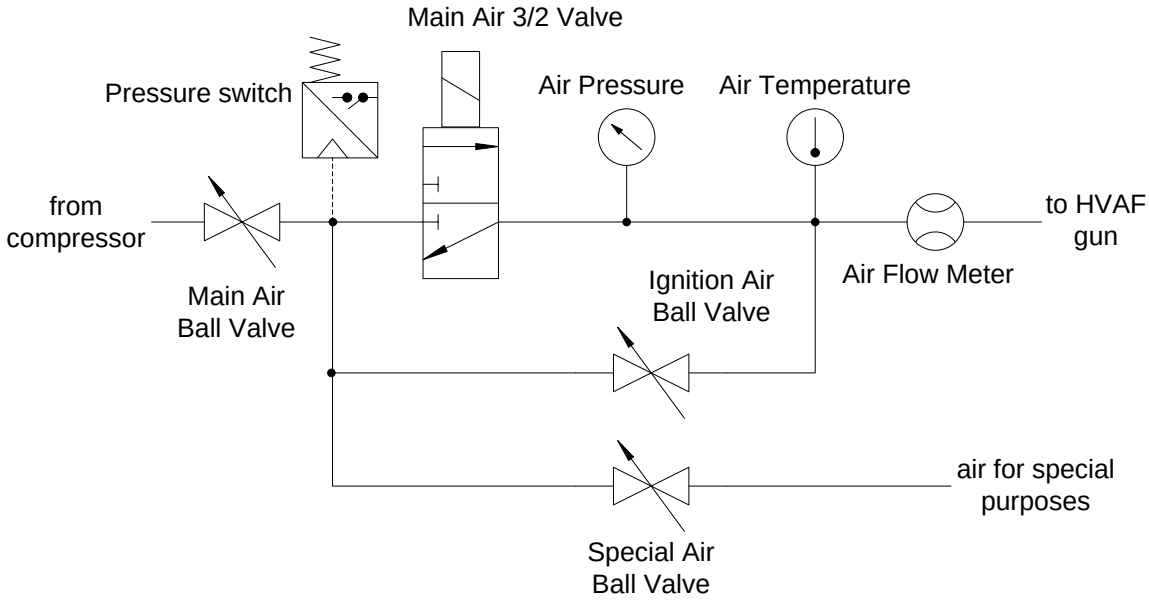


Figure 4.4: Schematic of the main air system

The fuel system is shown in Figure 4.5. The control system is equipped with a 20 l fuel tank that allows the system to operate for approximately 3 hours without refilling. The fuel is pressurised by a fuel pump. Fuel flow rate and fuel pressure are monitored by sensors. The fuel system is split after the fuel flow meter in two lines. The first line supplies a flame glow plug that is used to start the process. This glow

plug fuel line can be switched off by a 3/2 valve. On the glow plug fuel line is also a pressure regulator to limit pressure of the fuel that supplies the glow plug. The second line is the main fuel line and can also be switched off by a 3/2 valve. Additionally a proportional valve is in the main fuel line to adjust the amount of fuel that is supplied via this line. The fuel pressure sensor is located on the main fuel line between the proportional valve and the 3/2 valve. To prevent burning fuel from being pushed back into the control system, both fuel lines are equipped with non return valves.

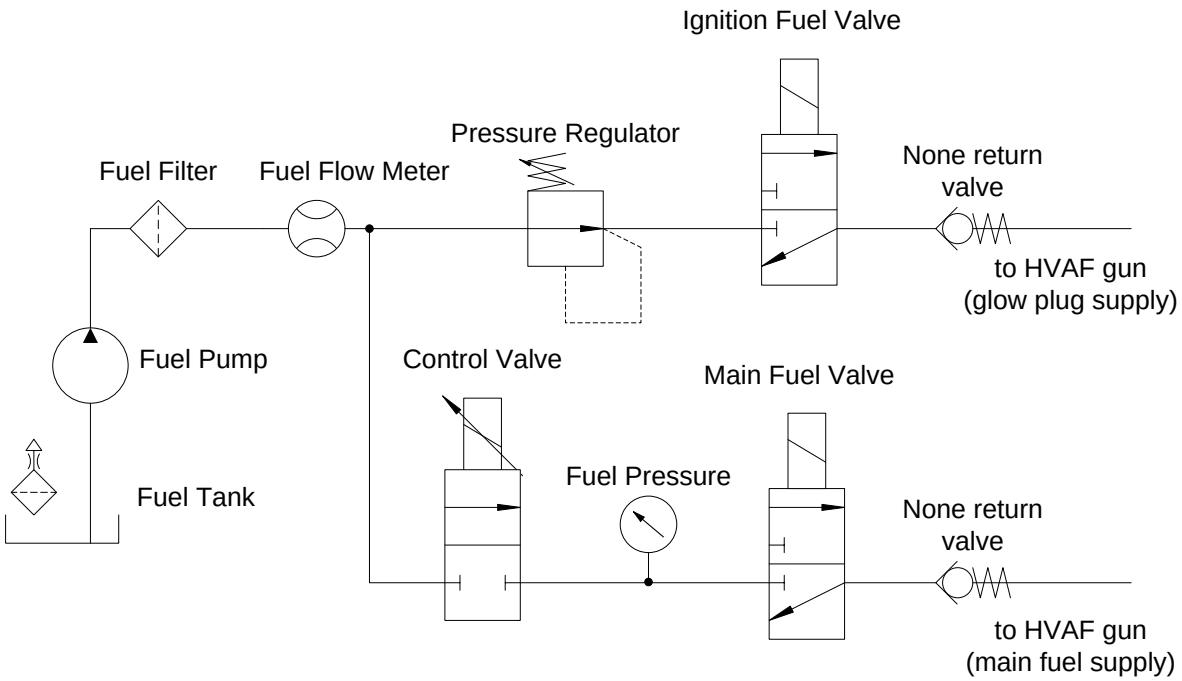


Figure 4.5: Schematic of the fuel system

The air system for the powder feeder is shown in Figure 4.6. It consists of an air dryer, an air flow meter, an air pressure sensor and the powder feeder itself. The Components are assembled in series.

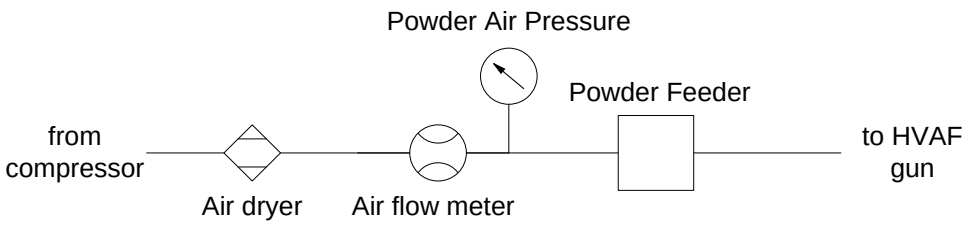


Figure 4.6: Schematic of the air system for the powder feeder

4.4.2 Electrical System

The heart of the electrical system is the PIC24H microcontroller on the multi I/O board. To make this microcontroller suitable for a control panel environment, custom built interface boards are used. On these interface boards, optocouplers for the digital inputs and relays for the digital outputs provide galvanically isolation.

The schematic of one input channel on the digital input board is shown in Figure 4.7. The Zener diode ZPD24 is used for over voltage and reverse voltage protection.

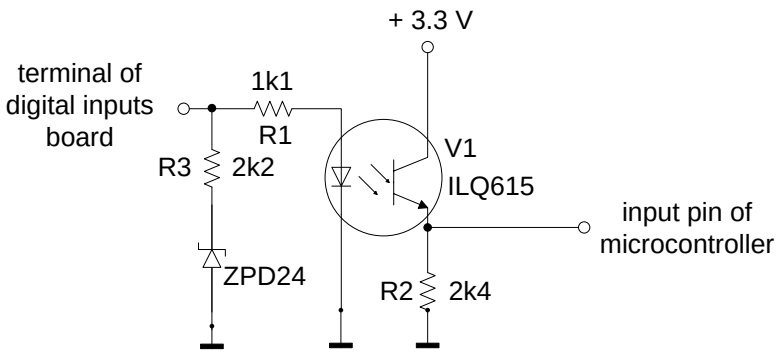


Figure 4.7: Schematic of one digital input channel

The schematic of one output channel on the digital output board is shown in Figure 4.8. The resistor R1 on the gate of the MOSFET V2 is incorporated to force a current flow from gate to ground, if the microcontroller pin is on high level. Since it was observed that due to disturbances with low energy the MOSFET was switched on, although the microcontroller pin was on ground potential. The calculation of the required current is shown in equation (4.5).

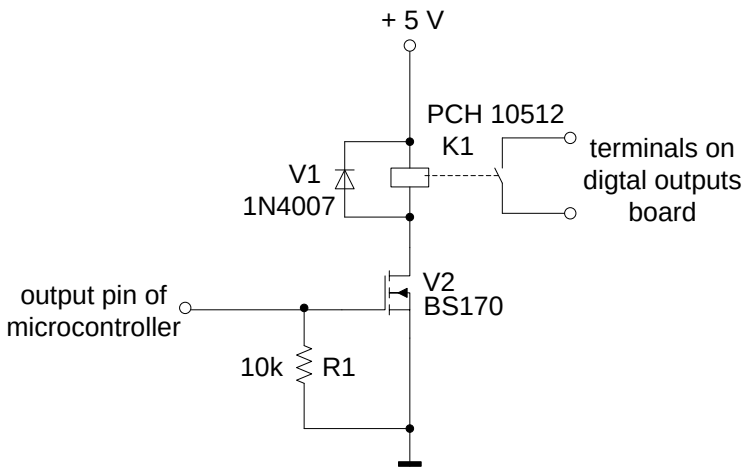


Figure 4.8: Schematic of one digital output channel

$$I_{out} = \frac{U_{out}}{R1} = \frac{3.3\ V}{10\ k\Omega} = 330\ \mu A \tag{4.5}$$

In a first version ribbon cable was used to connect the digital input and output boards to the multi I/O board. This led to electromagnetic interferences therefore all ribbon cables have been replaced by shielded cables.

For the high power loads, namely glow plug, fuel pump and powder feeder, additional relays are connected to the digital output board.

All used analog sensors have analog current outputs (4 – 20 mA). As the PIC microcontroller can only read analog voltages from 0 – 3.3 V the analog current signals must be converted to analog voltage signals. This conversion is achieved by measuring the voltage drops over high accurate resistors.

To provide a 4 – 20 mA analog current output, for controlling the proportional valve, a 16 bit Digital Analog Converter (DAC) with Serial Parallel Interface (SPI) is used.

For storing captured process data an external 32 Kbyte parallel SRAM is attached to the controller. Via RS232 serial interface, the controller can transfer the captured process data to a PC.

The ignition of the combustion process is done by means of a pilot flame obtained from an automotive flame glow plug. Therefore, the glow plug has its own 12V/30A

DC power supply. The other components in the control panel are supplied by an ordinary PC power supply and a 24V DC power supply.

A SysML use case diagram is applied to analyse what control elements are necessary for the operator to operate the system. This use case diagram is shown in Figure 4.9.

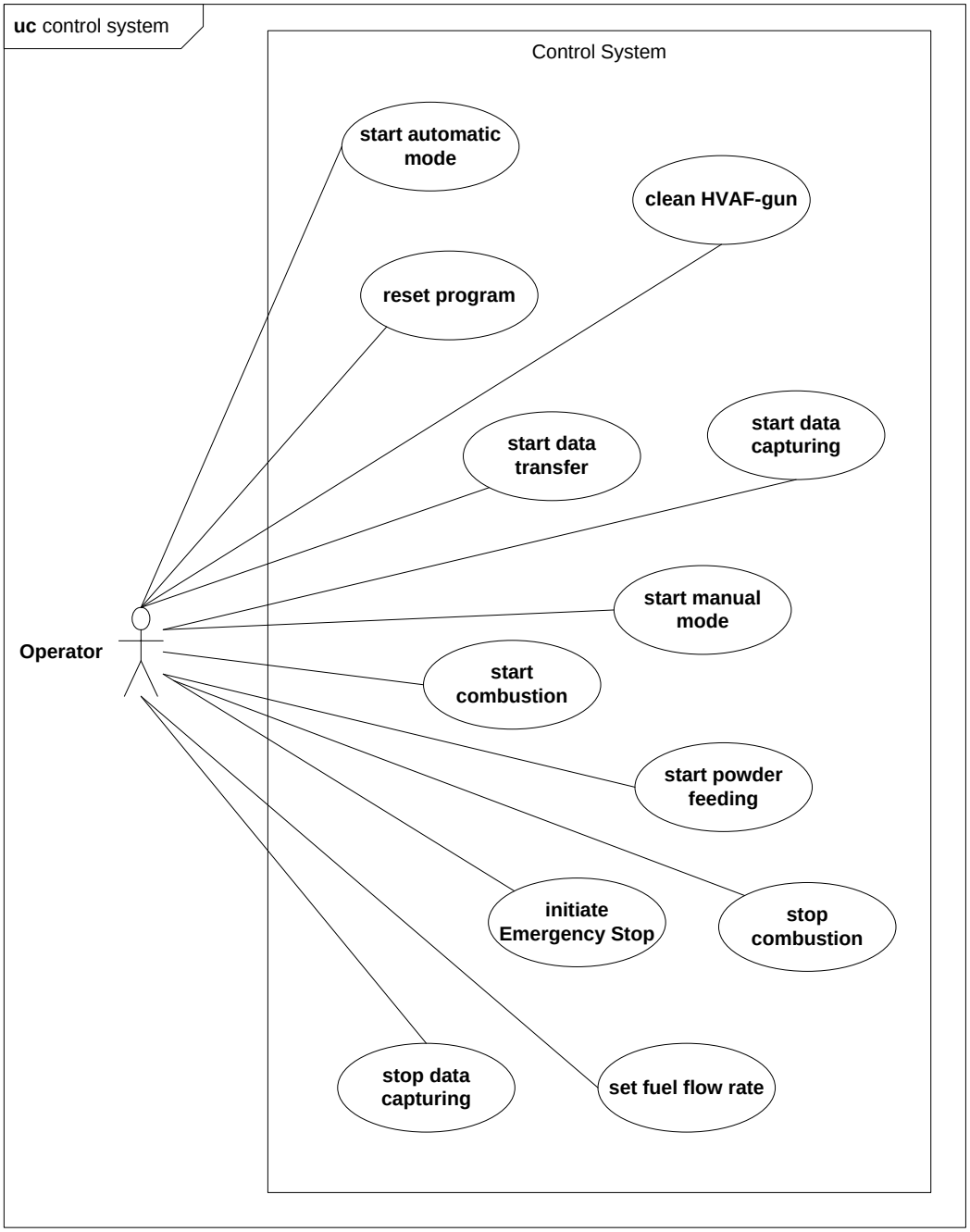


Figure 4.9: Use Case diagram for operating the controller

Twelve possible actions of the operator have been identified with the use case diagram. To serve these actions, one potentiometer and eight push buttons and switches are implemented in the user interface shown in Figure 4.10.

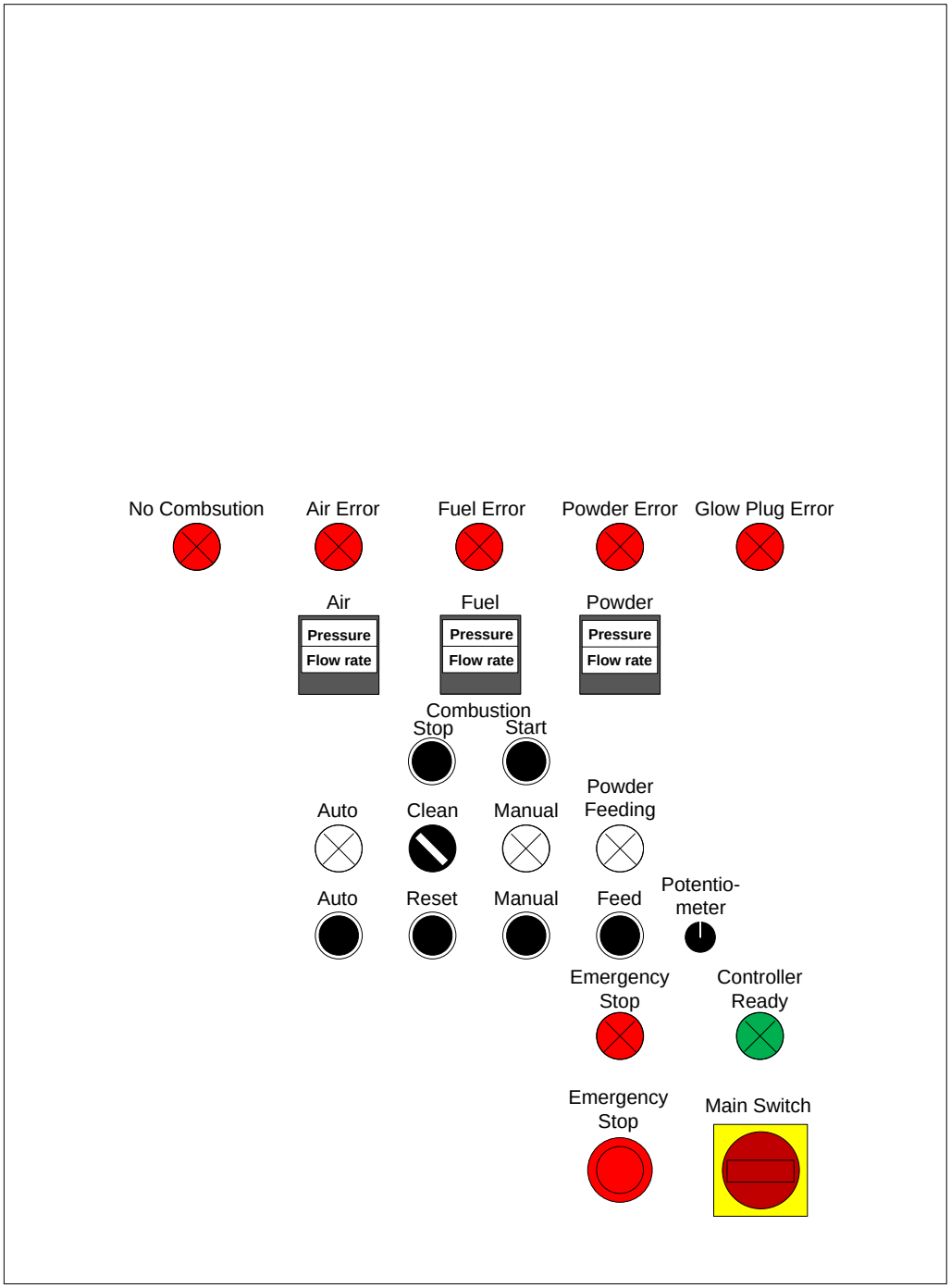


Figure 4.10: Layout of the control panel front

For displaying the process parameters (main air pressure and flow rate, fuel pressure and flow rate, powder pressure and flow rate) 3 numeric displays are used. Every

display can show two analog values and is connected in series to the measurement resistor of the interface board. This saves valuable calculation time and enables the programmer to focus on the actual control task.

The built control system for the HVAF thermal spray process is shown in Figure 4.11.

Figure 4.11: The control system for the HVAF thermal spraying process

4.5 Software Design and Implementation

The design and the implementation are described in this section. The software is designed as a framework for further applications.

First an overview over the structure of the software is given in 4.5.1 “Software Structure Overview”.

The protocol for data transfer between the microcontroller and a PC is designed in section 4.5.2 “Protocol for Serial Communication”.

The program for the microcontroller is described in section 4.5.3 “Microcontroller Program”.

For communication with the microcontroller a PC application is required which is described in section 4.5.4 “PC Application”.

4.5.1 Software Structure Overview

The developed software consists of two main packages, the “PC Application” and the “Microcontroller Program”. The UML package diagram of the software is shown in Figure 4.12.

The PC application consists of the two sub packages “Serial Communication” and “Graphical User Interface (GUI)”. As the name says, the graphical user interface of the PC application is realised in the package “Graphical User Interface (GUI)”. In the package “Serial Communication” the communication via RS232 between PC and microcontroller is implemented.

The microcontroller program is split into four sub packages, namely “Automatic Mode”, “Manual Mode”, “Data Transfer” and “Main Program”. The package “Main Program” is the main program of the microcontroller from which the other packages are called. The manual and automatic modes of the control system are coded in the packages “Manual Mode” and “Automatic Mode”, while the communication via RS232 between PC and microcontroller is implemented in the package “Data Transfer”.

In the following sections the several packages are described more detailed.

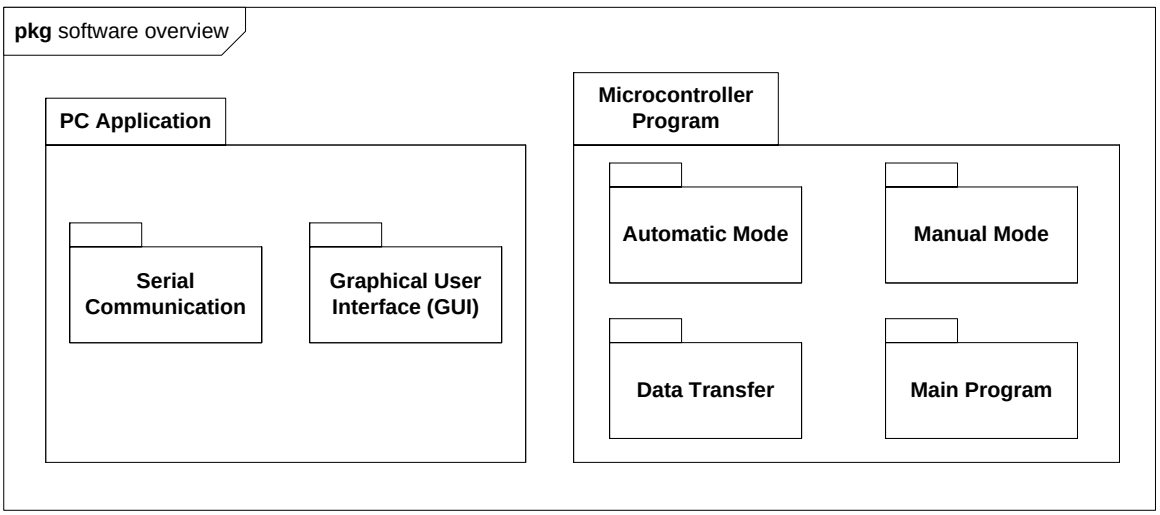


Figure 4.12: Package diagram of the software

4.5.2 Protocol for Serial Communication

The transfer of measurement data from the microcontroller to the PC is done via RS232 serial interface. The main reason for choosing RS232 is its simplicity of implementation with a microcontroller.

To detect loss or corruption of data during transmission, a simple protocol is defined inspired by Saal (23). The process data is packed into data packages that consist of the length of the data in bytes, the actual data and the 16 bit Cyclic Redundancy Check (CRC) of the data Figure 4.13.

Length (2 byte)	Data (16 to 48 byte)	CRC (2 byte)

Figure 4.13: Data package format

Each data package sent out by the microcontroller has to be acknowledged by the PC. If the PC receives the data package correctly it acknowledges with the hexadecimal value of 0xff otherwise with 0x00. The microcontroller program counts the number of high bits in the received byte. If this number is bigger than four, the data package was received correctly by the PC. Otherwise a transmission error occurred and the data package will be sent out again. The two hexadecimal values 0xff and 0x00 are chosen on purpose to increase the robustness of the communication against disturbances. As the chance that four or more bit errors occur is very small.

The UML sequence diagram of the serial communication process is shown in Figure 4.14. The communication is initiated by the microcontroller that sends out the message “controllerReady” and waits for the acknowledgement “pcReady” from the PC. If this was successful, the microcontroller sends out the first package which contains the number of packages that will follow. If the PC receives the package not correctly, the microcontroller sends out the package again. If this happens four times, the communication is aborted as there is most likely something wrong in the communication path. Otherwise the packages containing process data are send out one after each other by the microcontroller. For each package the microcontroller waits for an acknowledgement from the PC. Again, if the PC receives a package not correctly, this package is sent out again by the microcontroller. If the same package is received four times not correctly by the PC, the communication is aborted. If not, the microcontroller carries one with sending out the following packages until all packages have been sent out.

The specific implementation for the microcontroller is described in section 4.5.3.4 “Data Transfer” and for the PC in section 4.5.4.2 “Serial Communication”.

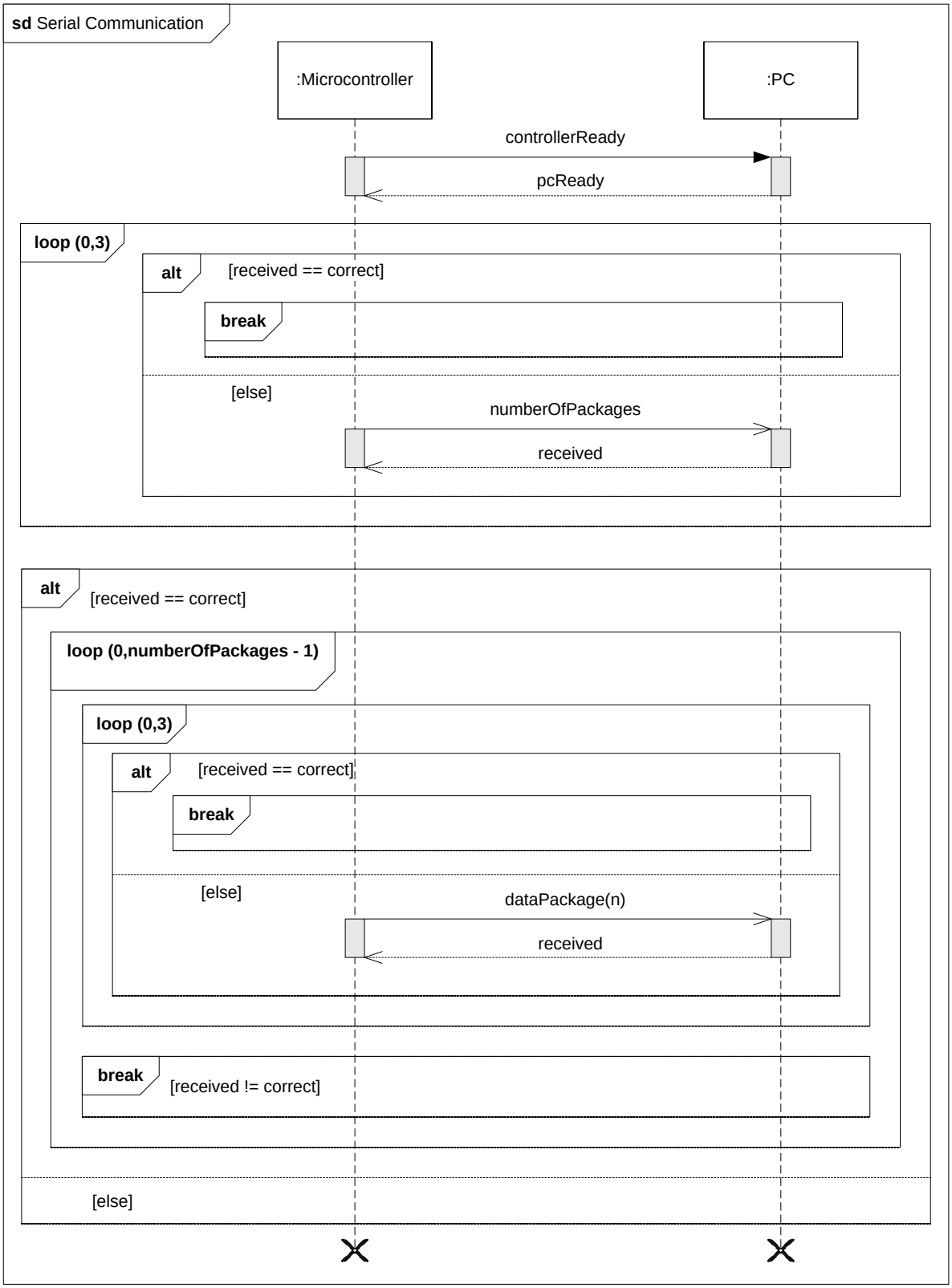


Figure 4.14: Sequence diagram of serial communication

4.5.3 Microcontroller Program

The software for the microcontroller is designed using Unified Modelling Language (UML) state charts, timing diagrams and activity diagrams. The actual program is written in C and is coded in the Microchip MPLAP Integrated Development Environment (IDE). To generate the binary file for the microcontroller the Microchip C30 C-compiler is used.

In section 4.5.3.1 “Main Program” the main program is described, from which all other programs are called.

To allow the user to run the control system in manually the manual mode is designed and implemented in section 4.5.3.2 “Manual Mode”.

The automatic mode is described in 4.5.3.3 “Automatic Mode”.

The implementation of the in section 4.5.2 “Protocol for Serial Communication” described protocol for serial communication is done in section 4.5.3.4 “Data Transfer”.

4.5.3.1 Main Program

The main program is designed as a finite state machine with seven states as shown in Figure 4.15. When the controller is switched on, the program starts in the state “Hardware Initialisation”. After completion of this state, the “Reset” state is entered. Now state transitions only occur when the corresponding digital inputs are set. That means if the program is in “Reset” state, pushing the “Auto” button leads into state “Automatic”, pushing the “Manual” button leads into state “Manual”, pushing the “Feed” button leads into state “Data Transfer”, pushing the “Emergency Stop” button leads into the state “Emergency Stop” and switching on the switch “Clean” leads into the state “Clean”. To get from the state “Clean” back into the state “Reset” the “Clean” switch must be switched off. For all other states except for “Hardware Initialisation” this is achieved by pushing the “Reset” button. Pushing the “Emergency Stop” button leads to a state transition into the state “Emergency Stop” regardless in which state the program actually is, except for the state “Hardware Initialisation”.

In the states “Emergency Stop” and “Reset” all digital outputs are switched off except for the “Emergency Stop” light in “Emergency Stop” and the “Controller Ready” light in “Reset”. This is done to clearly separate a normal stop from an emergency stop

and that the operator has to acknowledge an emergency stop before carrying on with normal operation. “Clean” offers a simple cleaning function for the spray gun by switching on the main air. The states “Manual”, “Automatic” and “Data Transfer” will be described more detailed in the following sections as they are more complex.

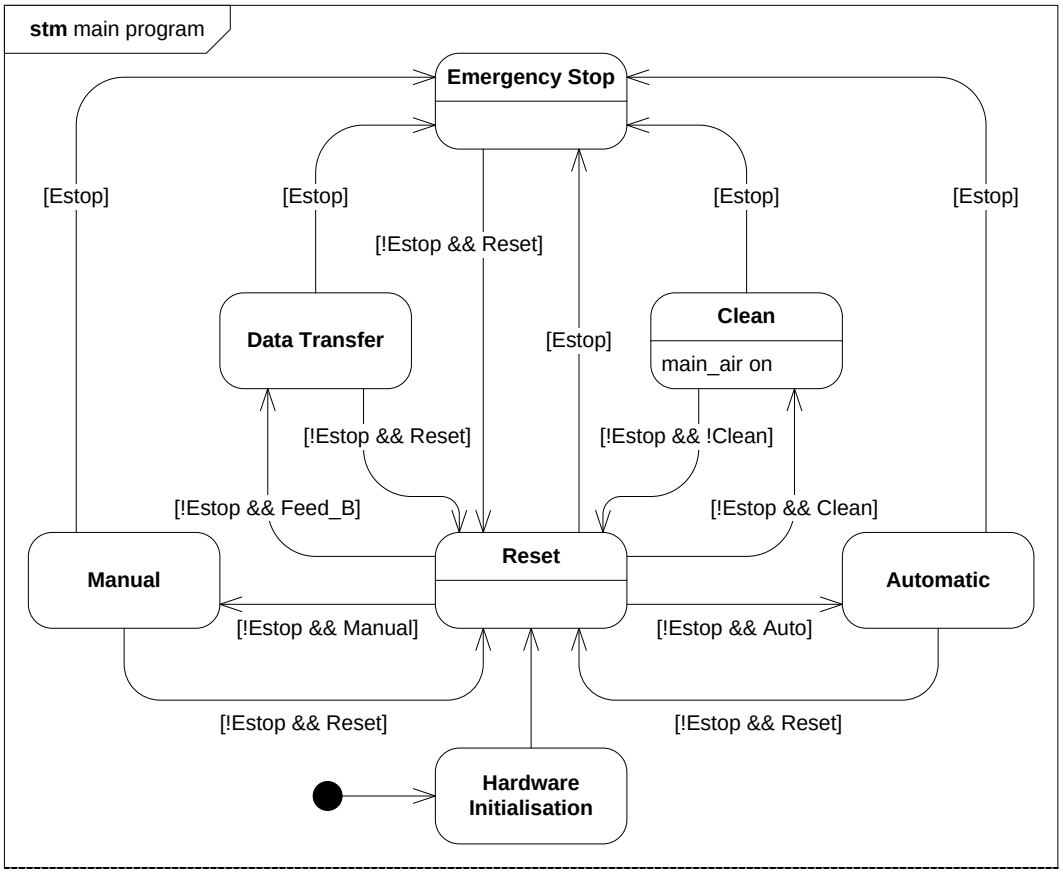


Figure 4.15: Main program

The hardware of the microcontroller and its peripherals is initialised in the “Hardware Initialisation” state. This includes initialisation of the watch dog timer, the oscillator, the Universal Asynchronous Receiver Transmitter (UART), Interrupts, Timer1, Timer3, Timer5, Serial Peripheral Interface (SPI), Direct Memory Access (DMA) and the two Analog Digital Converters (ADCs), ADC1 and ADC2.

ADC1 is used when the user wants to capture data for later processing. Therefore it is configured to sample four analog inputs at a time (simultaneous sampling). While ADC2 is configured to sample one analog input at a time (sequential sampling) and its sampled analog values are used by the program. The sampling options of the

ADC modules are shown in Figure 4.16. A more detailed description of the sampling options can be found in the PIC24 reference datasheet (24).

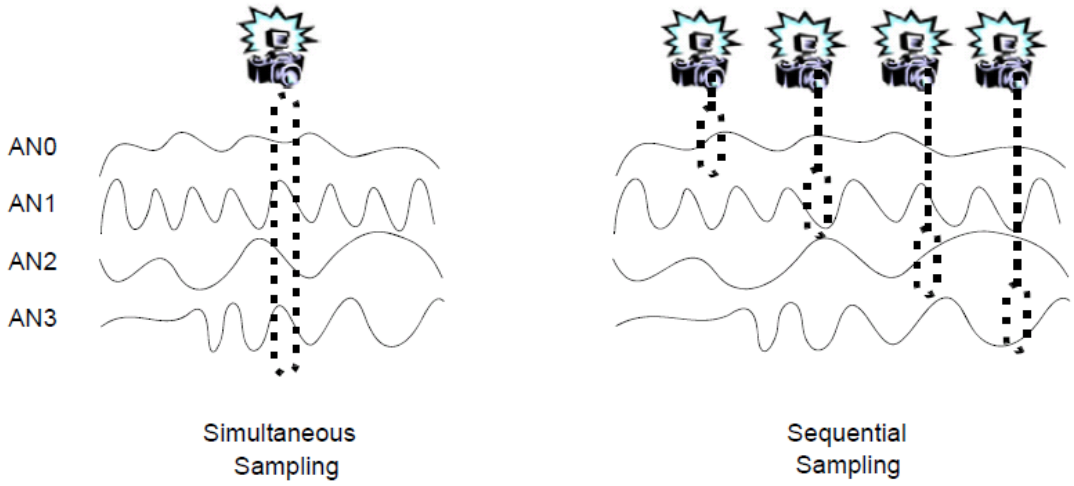


Figure 4.16: Sampling options of the Analog Digital Converter (ADC) modules (24)

4.5.3.2 Manual Mode

The main state “Manual” is modelled with two parallel finite state machines (Figure 4.17). One state machine is for starting, running and stopping of the process and the other is to activate and deactivate the Analog Digital Donverter 1 (ADC1), which enables the operator to capture data on the process at anytime as long as the program is in the main state “Manual”. However, a real parallel execution is not possible because only one processor is used. When the main state “Manual” is entered immediately the two sub states “ADC1 off” and “Ready” are entered.

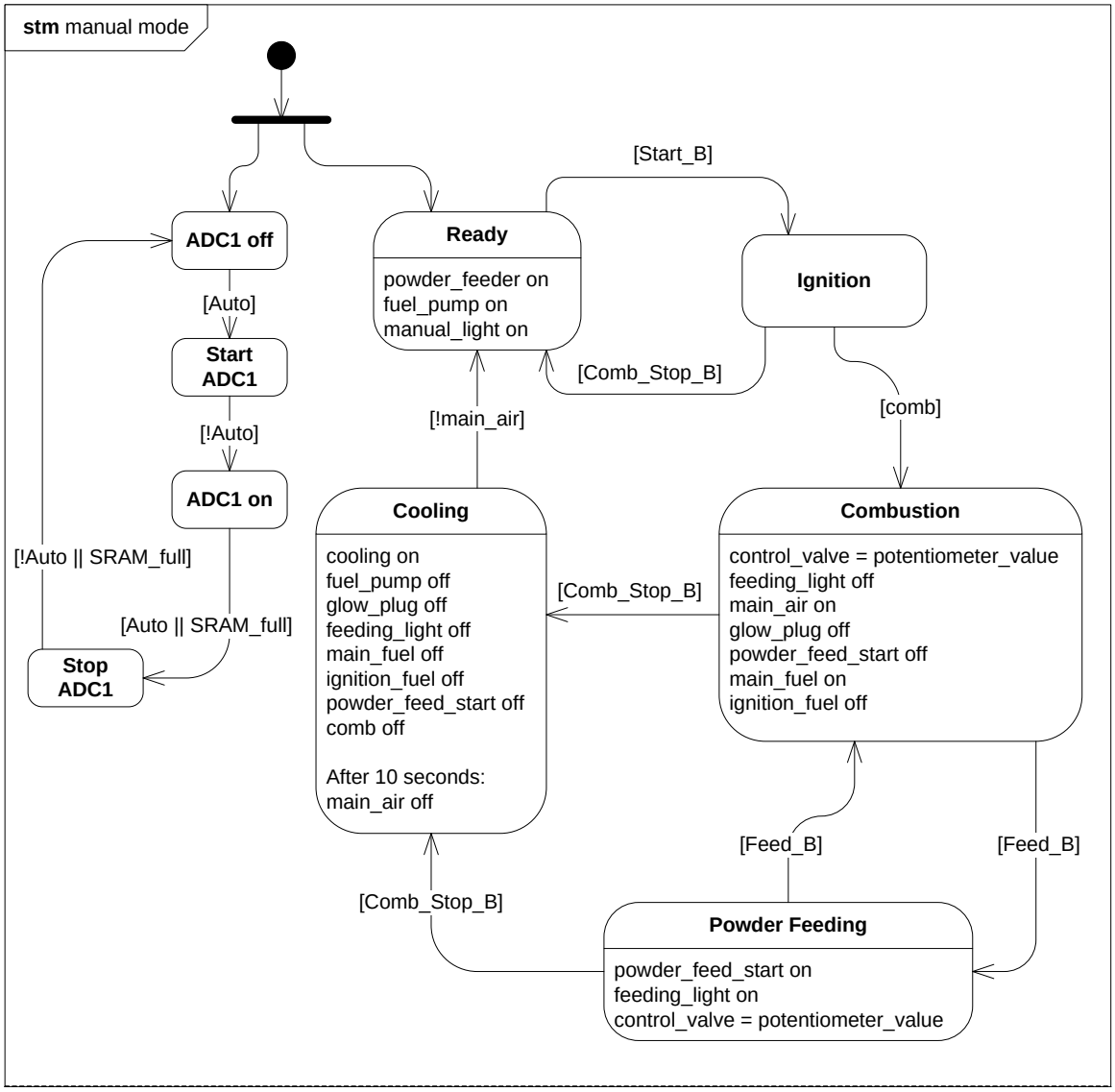


Figure 4.17: State diagram of manual mode

In the “Ready” state the powder feeder, the fuel pump and the “Manual” light are switched on. By pushing the “Combustion Start” button, the state “Ignition” is entered.

The “Ignition” state is also used in the automatic mode. It provides a time based ignition process. The timing diagram of ignition is shown in Figure 4.18. By entering the “Ignition” state the flame glow plug is switched on. After $t_i = 10\ s$ the fuel supply to the flame glow plug is switched on. At $t_i = 38\ s$ the flame glow plug is switched off and the main air is switched on. Two seconds later at $t_i = 40\ s$ the main fuel is switched on. After $t_i = 50\ s$ the fuel supply to the flame glow plug is switched off, the variable comb is set to true and the “Ignition” state is left.

In the manual mode the value of the potentiometer is captured every program cycle by the ADC2 and is written to the DAC via SPI. Hence the opening of the proportional valve is proportional to the position of the potentiometer the operator can regulate the flow of the main fuel.

In the automatic mode the proportional valve is opened via a ramp function from 0% open at $t_i = 40\ s$ to 85% open at $t_i = 50\ s$.

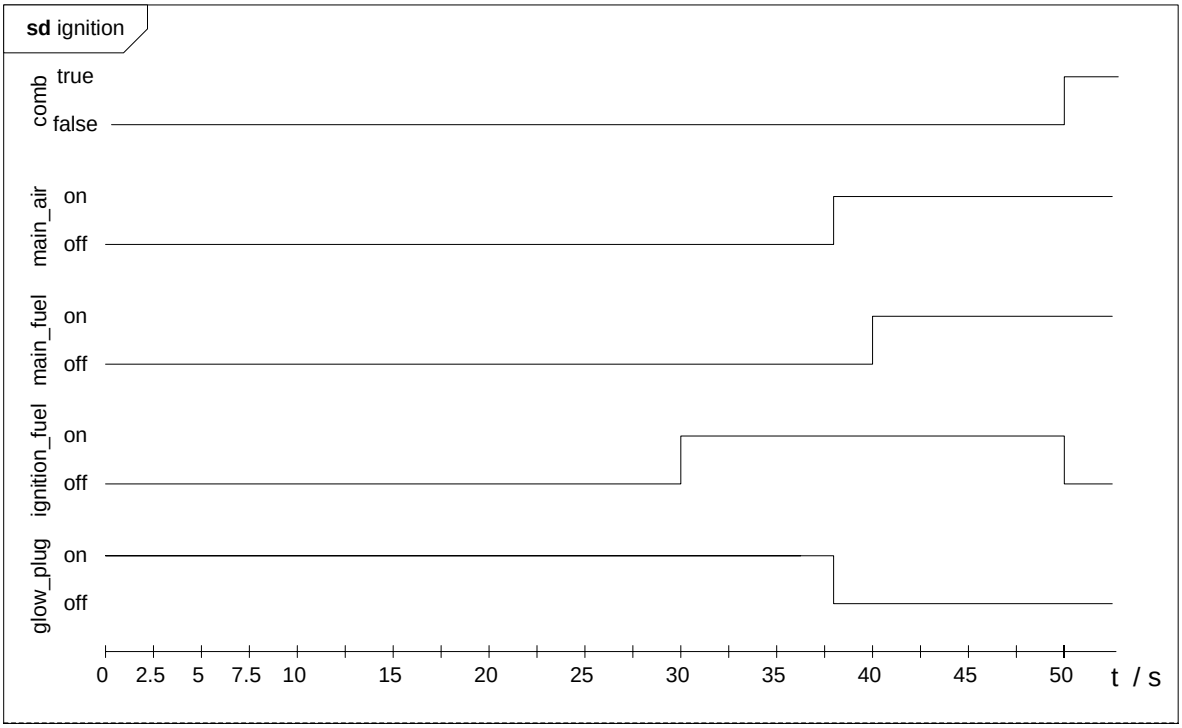


Figure 4.18: Timing diagram of ignition

The ignition can be interrupted by pushing the “Combustion Stop” button, which leads into the state “Ready” again.

If the ignition is not interrupted, the state “Combustion” is entered after the combustion is self sustaining. In the state “Combustion”, the main air and the main fuel stay switched on to keep the combustion going. Also in this state the opening of the proportional valve is proportional to the position of the potentiometer. So the operator can set the operating point of combustion by adjusting the position of the potentiometer. The feeding light, the glow plug, the start feeding contact and the ignition fuel are switched off in the state “Combustion”.

By pushing the “Feed” button the state “Powder Feeding” is entered. The outputs stay the same as in “Combustion”, except for the feeding light and the start feeding contact that are switched on. The powder feeder now supplies powder to the HVAF gun, so the actual thermal spraying can be done. If the “Feed” button is pushed again the state changes back to “Combustion”.

Both states “Combustion” and “Powder Feeding” can be interrupted by pressing the “Combustion Stop” button, which leads into the state “Cooling”. In the state “Cooling” the fuel pump, the glow plug, the feeding light, the main fuel, the ignition fuel and the start feeding contact are switched off. The only output that stays switched on is the main air to cool the HVAF gun down. After $t_c = 10\ s$ the main air is switched off and a state transition occurs into the state “Ready”.

The second state machine works independent from the above described. By pushing the “Auto” button the ADC 1 is started. When the ADC1 is running it captures the values of the analog inputs and these values are transferred via direct memory access (DMA) into the RAM of the PIC. The DMA is configured to operate in “ping pong mode”, see Figure 4.19. There are two DMA buffers situated in the RAM of the PIC. When the first buffer is full an interrupt is generated and the data in the buffer is written into the external SRAM. In the meantime the DMA transfers the captured values from the ADC1 into the second buffer in the RAM of the PIC. When this second buffer is full an interrupt is generated again and the data in the buffer is written into the external SRAM. In the meantime the DMA switches to the first buffer again. This carries on until the user stops the ADC1 by pushing the “Auto” button or until the external SRAM is full. The states “ADC1 off” and “ADC1 on” do not have a direct influence on the operation of the ADC1. The task of these states is that the user has to release the “Auto” button after the ADC1 has been started or stopped before a new operation is initiated.

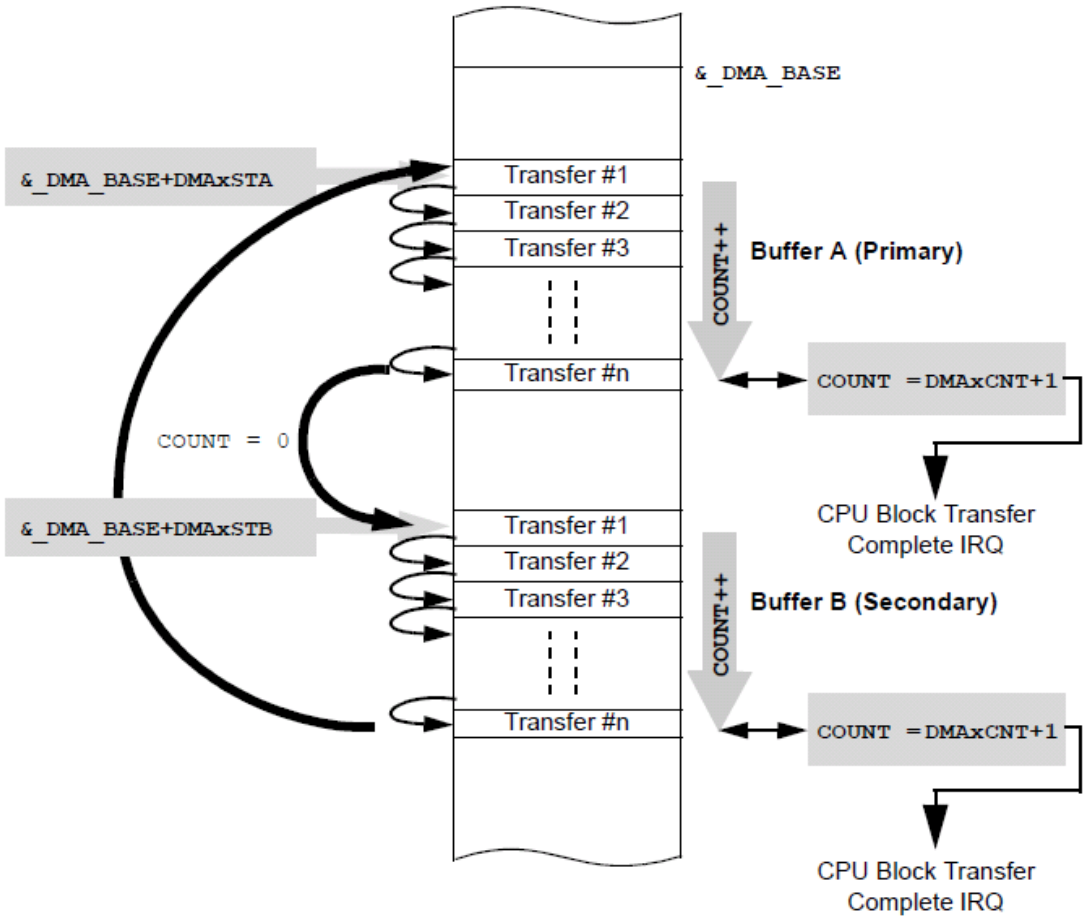


Figure 4.19: Buffer filling of the Direct Memory Access (DMA) module in “ping pong” mode (24)

4.5.3.3 Automatic Mode

In the automatic mode a state machine is used that is similar to the one for the control of combustion in manual mode. The state diagram of the automatic mode is shown in Figure 4.20.

When the automatic mode is called, the state “Testing” is entered. In “Testing” the main air is switched on for $t_t = 5\,s$ and the average air pressure is captured and saved in the variable “air_press”.

If the average air pressure is below $p_{air} = 5.5\,bar$ the air supply is not sufficient for the operation of the HVAF gun and therefore the state “Air Error” is called in which the air error light is switched on.

If the average air pressure is above or equal to $p_{air} = 5.5\,bar$ the state “Ready” is entered which is basically the same as in manual mode.

A state transition into the state “Ignition” only occurs if the “Combustion Start” button is pressed and the pressure of the air supply of the powder feeder is above or equal to $p_{powder} = 6.5\ bar$. If the pressure of the air supply of the powder feeder drops below $p_{powder} = 6.5\ bar$ during ignition the ignition is interrupted. The powder error light is switched on and the state “Cooling” is called. The same happens if in the states “Combustion” and “Powder Feeding” the pressure of the air supply of the powder feeder drops below $p_{powder} = 6.5\ bar$. This is done to prevent flames of being pushed into the powder supply line because of to low pressure.

The states “Combustion” and “Powder Feeding” are basically the same as in the manual mode except for controlling the opening of the proportional valve. While in the manual mode the opening of the proportional valve is proportional to the position of the potentiometer, the opening of the proportional valve is calculated from the sensor values in automatic mode. The state transitions from and to these states stays the same as in manual mode, except for the additional conditions described above.

Also the state “Cooling” stays the same as in manual mode. After the main air has been switched off and the powder error light is on, the state “Powder Error” is called, otherwise the state “Ready” is called.

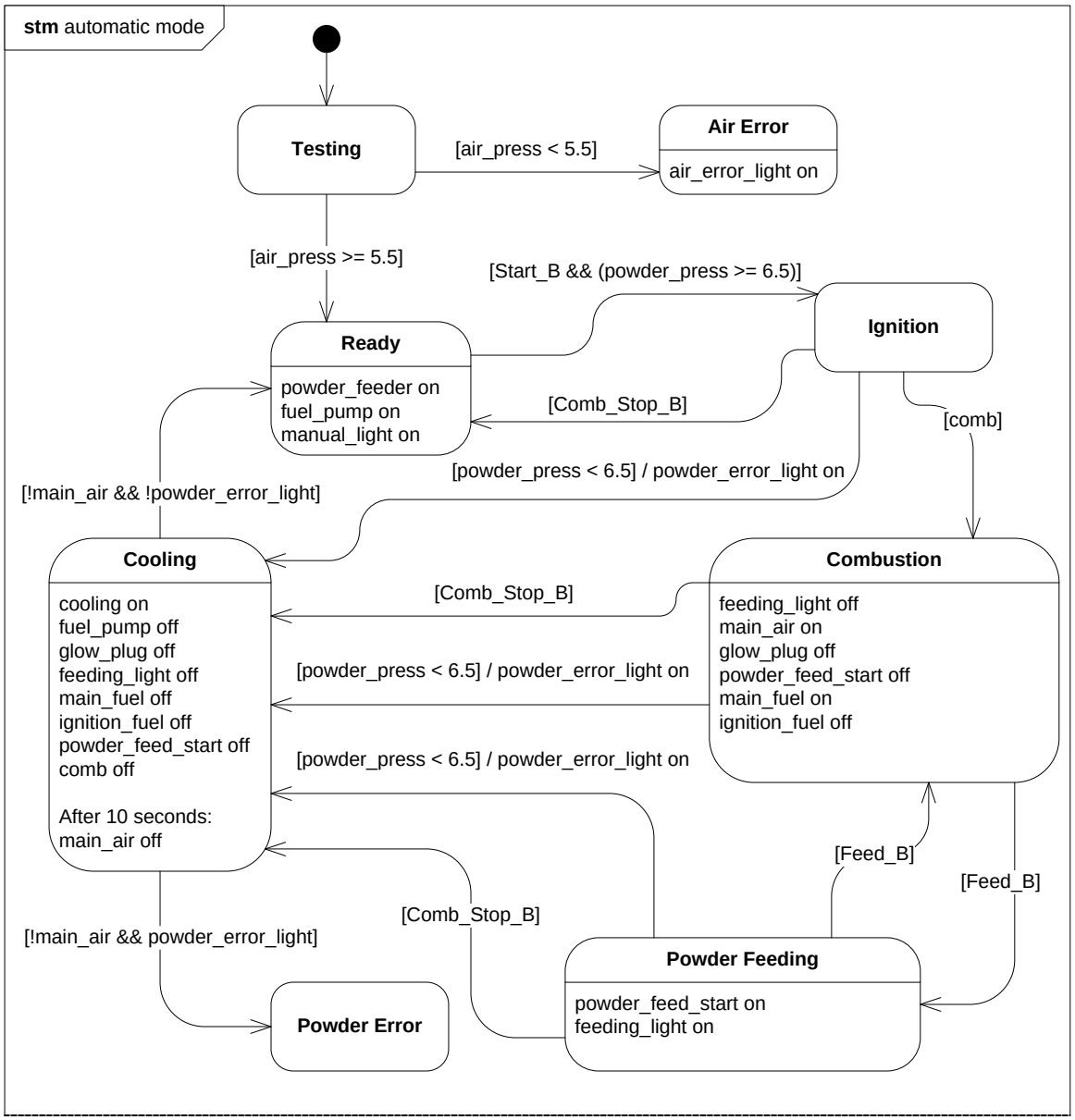


Figure 4.20: State diagram automatic mode

4.5.3.4 Data Transfer

The transfer of the captured process data via RS232 to a PC for further processing is implemented in the state “Data Transfer”. Figure 4.21 shows the activity diagram of the main state “Data Transfer”.

A list of variables used for the control of the program flow is shown in Table 4.3. Because the Microchip C30 compiler those not support Boolean variables, these variables are realised as integer variables like in traditional C, where a value of zero is corresponding to false and a value different from zero is corresponding to true. In the following description only the values true and false are used for these variables.

Variable name	Description	Data type
error_counter	Number of transmission error of the current package	unsigned int
stack_position	Actual address of the SRAM from which data is read	unsigned int
end_position	Last address of the SRAM on which data is stored	unsigned int
inicomp	Initialisation complete	int
ini	No initialisation done yet	int
err	A transmission error occurred, the data must be send out again	int
datacorrect	The PC received the data correctly	int
datareceived	Data is in the receive buffer of the UART	int

Table 4.3: Variables in Data Transfer

After initialisation of the control variables, the character ‘>’ is send out via UART, to signal that the microcontroller is ready for the transfer of data. The timeout timer is reset and the program waits for *datareceived* to change to true, which is done by an interrupt if data arrives at the receive buffer of the UART. If after $t_{out} = 10s$ no data has arrived at the receive buffer of the UART state “Data Transfer” is left and therefore the serial communication is ended.

Otherwise data has arrived and is then checked if it has either the hexadecimal value of “0xff” or “0x00”. If the hexadecimal value is “0xff” the PC has received the previously sent out data correctly, *datacorrect* is set to true. If the hexadecimal value is “0x00” the PC has received the previously sent out data not correctly, *datacorrect* is set to false.

In the case that *datacorrect* is false, *err* is set to true and *error_counter* is incremented. If the value of *error_counter* is equal to or above four, the communication is ended. If the value of *error_counter* is below four, ‘x’ is send out, the timeout timer is reset and the controller waits for data to arrive at the receive buffer of the UART.

In the case that *datacorrect* is true, *error_counter* is set to zero. To determine if the initialisation of the communication is complete, *inicomp* is checked.

If the initialisation is complete, *inicomp* is true. It is then checked whether a transmission error occurred or not. If a transmission error occurred *err* is set to false, data is read at the address *stack_position* from the SRAM and written into *buffer*. The

further processing of *buffer* is described later on, as the processing of *buffer* is for all other possibilities the same. If no transmission error occurred *stack_position* is incremented. If *stack_position* is smaller than *end_position*, data is read out from the address *stack_position* of the SRAM and written into *buffer*. If *stack_position* is bigger or equal to *end_position*, all data has been transmitted successfully and therefore serial communication is ended.

If the initialisation is not complete yet, *inicomp* has the value false. In this case it is checked if an initialisation was already performed or not.

If no initialisation was performed yet, *ini* is true. The variable *ini* is then set to false and the number of data packages is calculated and written into *buffer*.

If an initialisation was performed already, *ini* is false and *err* is checked whether a transmission error occurred or not. If a transmission error occurred, *err* is set to false, the number of data packages is calculated and written into *buffer*. If no transmission error occurred, *inicomp* is set to true, data is read from the address *stack_position* of the SRAM and written into *buffer*.

After data has been written into *buffer*, the length of buffer is calculated. The length and buffer are then written into *package*. From *package* the CRC is calculate and is then attached to *package*. The CRC calculation is done by crc.c written by Barr and Massa (25). The variable *package* is then sent out via UART and *datareceived* is set to false. The timeout timer is reset and it is waited until new data arrives.

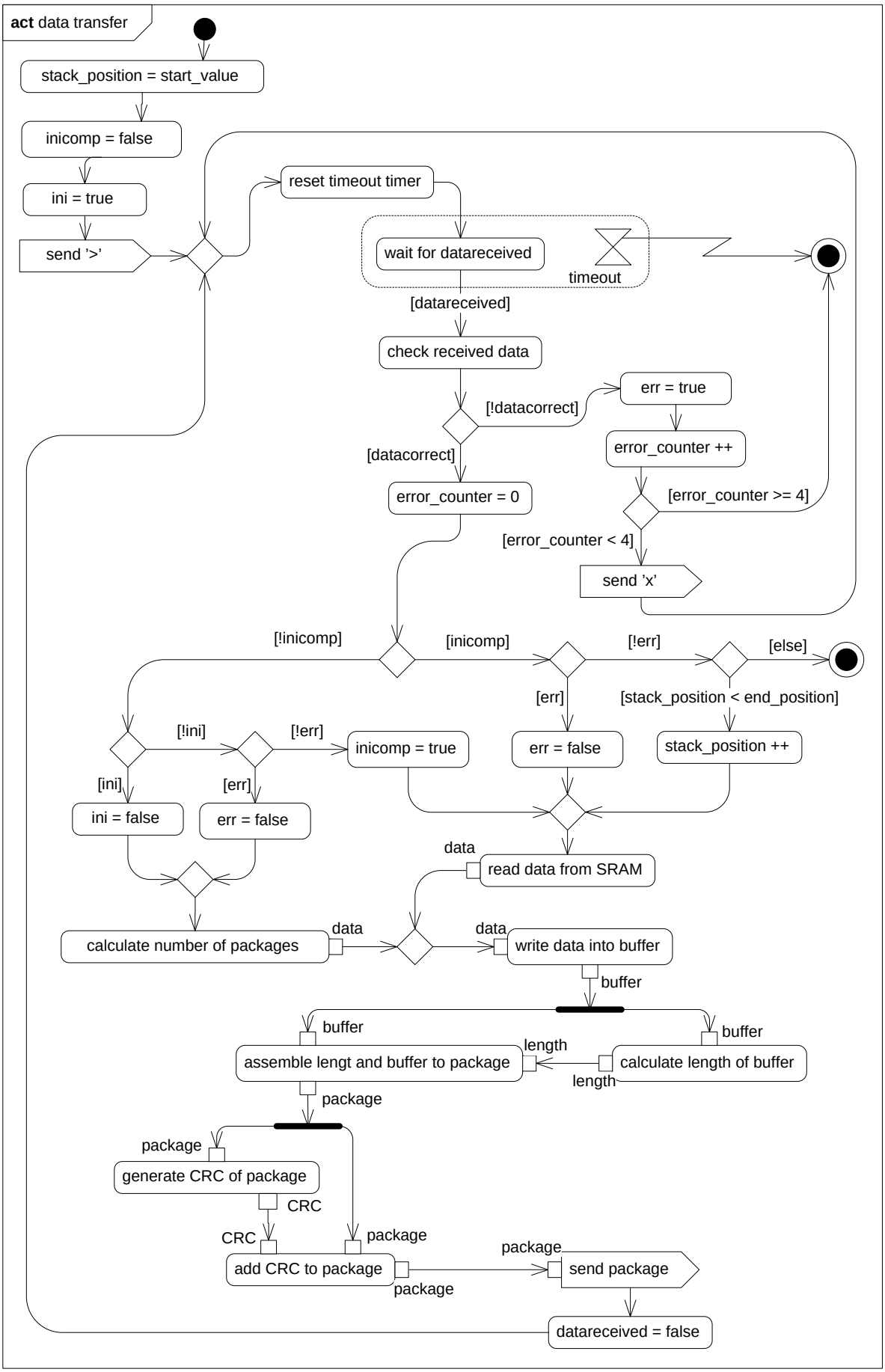


Figure 4.21: Activity diagram of data transfer

4.5.4 PC Application

The PC application is designed using UML use case and activity diagrams. The actual program is written in C# and is coded in the Microsoft Visual Studio development environment using Microsoft .Net Framework.

The graphical user interface is designed in section 4.5.4.1 “Graphical User Interface”.

The implementation of the in section 4.5.2 “Protocol for Serial Communication” described protocol for serial communication is done in section 4.5.4.2 “Serial Communication”.

4.5.4.1 Graphical User Interface

For the design of the graphical user interface, the use cases of the PC application are modelled, see Figure 4.22. The operator needs to select the serial port to which the control system is connected. Furthermore the operator must connect to the control system to start the transfer of data. Finally the operator must have the possibility to save the received data.

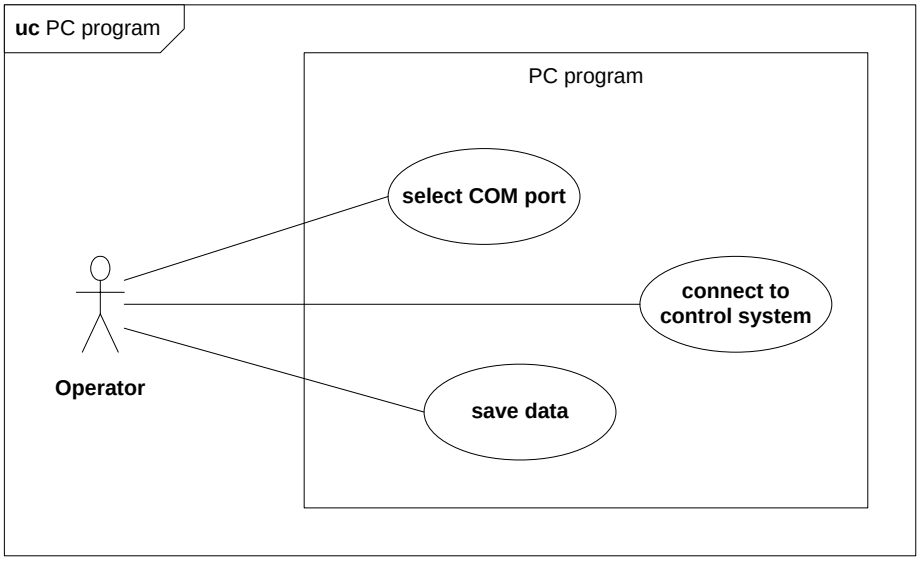


Figure 4.22: Use case diagram of the PC application

According to these use cases the graphical user interface is designed. For every use case a control element is implemented. The GUI is shown in Figure 4.23. For the use case “select COM port” a dropdown list is used, for “connect to control system” and for “save data” buttons are used. Additionally a status label and a progress bar are

implemented to give the operator a feedback on the status of the communication. Furthermore a textbox is implemented in which the received data is displayed.

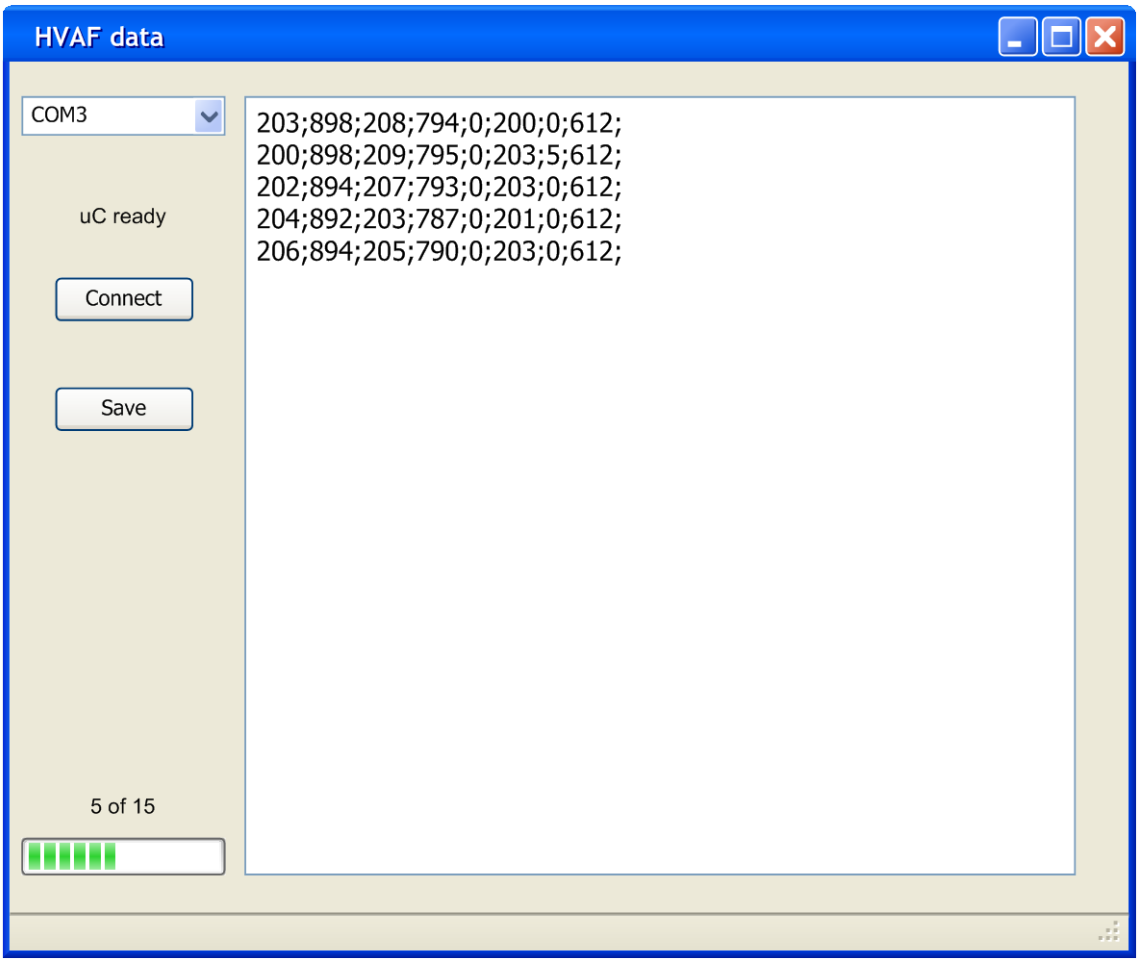


Figure 4.23: Graphical User Interface (GUI) of the PC application

For the use case “select COM port” the activity diagram is shown in Figure 4.24. First the available serial ports are displayed. The user has then the possibility to choose one of the available serial ports. If the selected port is open, it is tried to close the port. Now it is tried to open the selected serial port. If this is possible the serial port can now be used by the application. If not, a error dialog is displayed to inform the user that the application was not able to open the selected serial port.

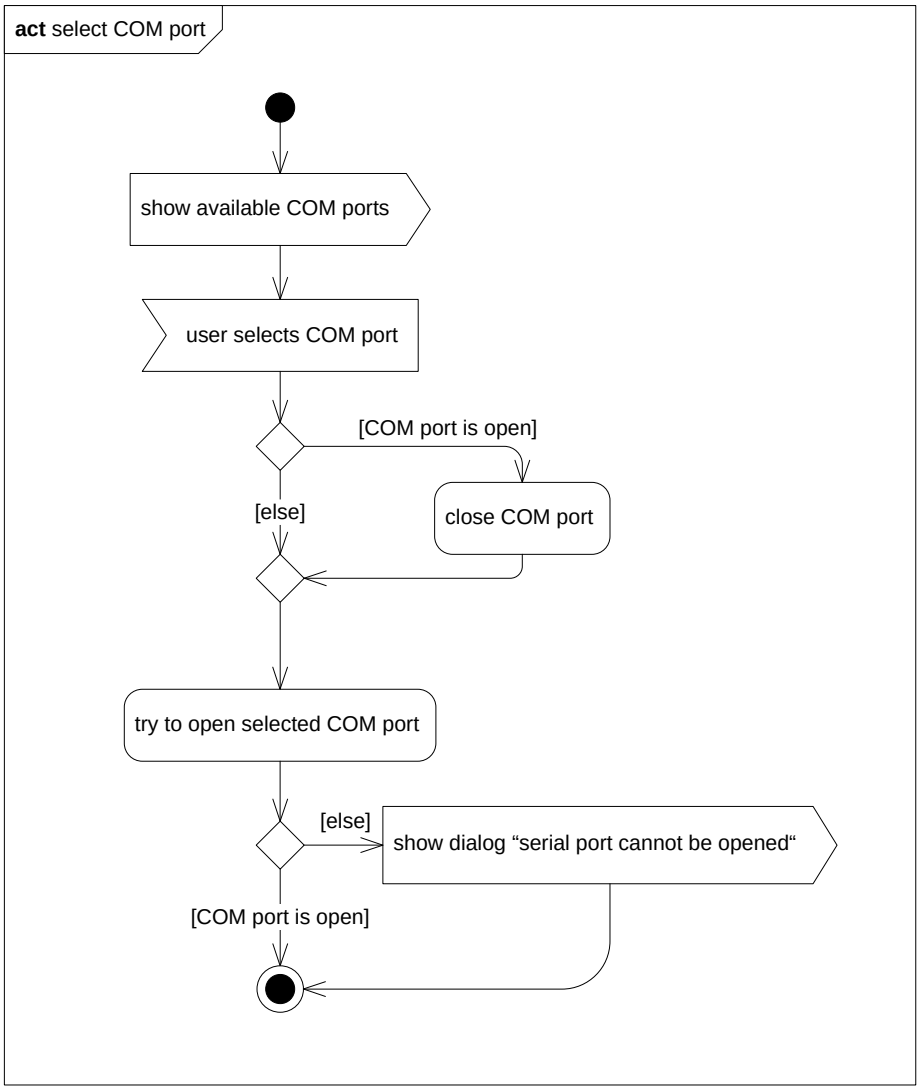


Figure 4.24: Activity diagram for use case select COM port

The activity diagram for the use case “save data” is shown in Figure 4.25 and is pretty simple. The .Net Framework supplies already a save dialog, so this dialog is used to save the data in the textbox. The save dialog is configured to save the data as .txt file by default. There are two reasons why the .txt format was chosen. One is simplicity, because the data in the textbox is already in the right format and the other is compatibility, because .txt files can be read by numerous programs for example Microsoft Excel or Matlab/Simulink from The MathWorks.

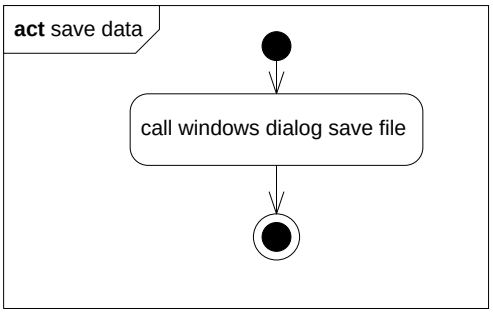


Figure 4.25: Activity diagram for use case save data

The activity diagram of the use case “connect to control system” is shown in Figure 4.26. When the activity is called, the text in the textbox is cleared. If the selected serial port is closed a dialog is displayed, that this serial port is closed. Otherwise the data in the inbuffer of the serial port is discarded, timer1 and timer2 are stopped, a initial value is assigned to the variables *progressBar.Value*, *lock1*, *lock2*, *anz* and *inicomplete*. Finally “0xff” is sent out via the serial port to signal the microcontroller that the application is ready.

What timer1, timer2 and the variables *progressBar.Value*, *lock1*, *lock2*, *anz* and *inicomplete* are used for, is described in the following section 4.5.4.2 Serial Communication.

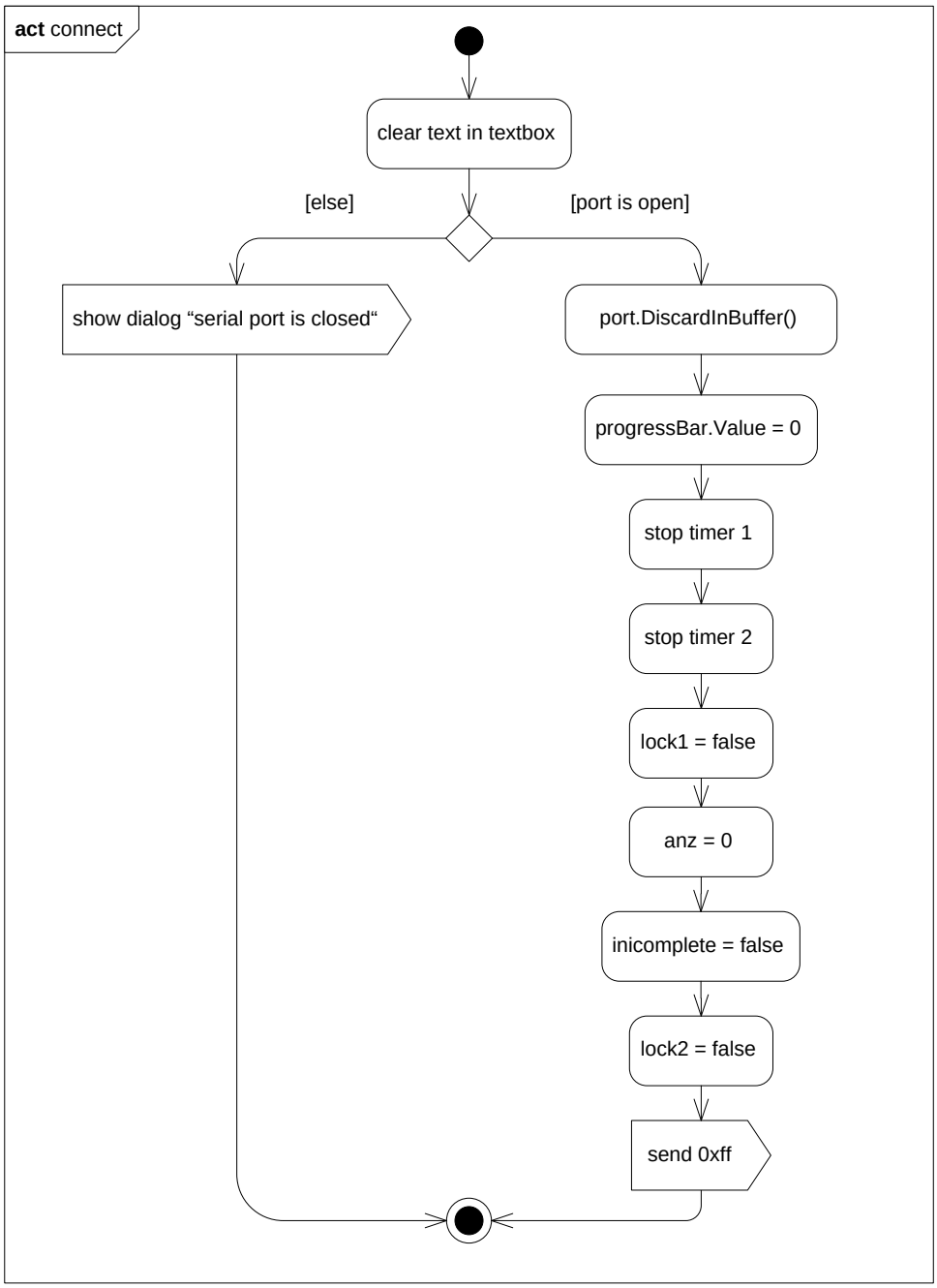


Figure 4.26: Activity diagram for use case connect to control system

4.5.4.2 Serial Communication

For serial communication the class SerialPort is supplied by the Microsoft .Net Framework. With the properties of this class the serial port of the PC can be configured. Also the class SerialPort contains several methods and events that are useful for serial communication. One of these events is the DataReceived event. Because “the DataReceived event is not guaranteed to be raised for every byte

received” (26) this event needs special treatment. Two timers, namely timer1 and timer2 are used to compensate the lack of reliable data reception detection.

After the communication with the microcontroller is established, timer1 is used to check the in-buffer of the serial port if the first two bytes of a data package have already arrived after the corresponding time. These two bytes represent the length of the data package. From the length of the data package the transmission time for the rest of the data package is calculate and timer2 is set according to this time. When timer2 has elapsed it is checked if the complete package has arrived.

A list of the variables used in serial communication is shown in Table 4.4.

Variable name	Description	Data type
lock1	Connection to microcontroller is established	bool
lock2	DataReceived has occurred at least once for the current package	bool
inicomplete	The initialisation of communication is complete	bool
err	Indicates transmission error	bool
crc	Extracted CRC from data package	int
check	Calculated CRC of the data in the data package	int
pack	Total number of data packages	int
anz	Number of received data packages	int
len	Length of the current data package	int

Table 4.4: Variables for Serial Communication

The activity diagram of the method that is called when a DataReceived event occurs is shown in Figure 4.27. When the method is called, the data from the in-buffer of the serial port is written into *inputData*. In case that *inputData* is empty the method is left. Otherwise, by looking at *lock1*, it is checked if the connection to the microcontroller is already established.

If the connection is established, it is checked if *lock2* is true and therefore timer1 has been started. If *lock2* is false, it is set to true and timer1 is started with an elapse value of 20 times the bit time. This is the transfer time for two bytes of data, since the serial port is configured for working with 1 start bit, 8 data bits and 1 stop bit. Afterwards *buff* and *inputdata* are merged and written into *buff*. The method is then

left. If *lock2* is true, *buff* and *inputdata* are merged and written into *buff* and the method is left without any other activity.

If the connection is not established, it is checked if the value of *inputData* is equal to the character ‘>’ which corresponds to microcontroller ready. If the microcontroller is ready, the status label is set to “uC ready” and *lock1* is set to true. The connection to the microcontroller is now established. Afterwards the same sequence is carried out as it is done if the microcontroller is not ready. The data in the in-buffer of the serial port is discarded, *buff* and *inputData* are emptied and the method is left.

The activity diagram of the timer1 elapsed event is shown in Figure 4.28. If timer1 has elapsed, this event is called. The timer1 is stopped and *buff* and the data in the in-buffer of the serial port are merged and assigned to *text*. It is then checked if a transmission error occurred previously.

If *err* is true, a transmission error occurred previously. The variable *lock2* is set to false, *buff* and *inputData* are emptied. If *text* equals to ‘x’ the microcontroller has acknowledged the transmission error, *err* is set to false, “0xff” is sent out via the serial port and the method is left. Otherwise “0x00” is sent out via the serial port and the method is left.

If err is true, the normal procedure is carried out. It is checked if enough data has arrived already, to extract the length of the current data package. Because the first two bytes of a data package contain the length, *text* must at least contain two bytes.

If *text* contains two or more bytes, the first two bytes of *text* are extracted and it is tried to convert them to integer. If the conversion is possible the integer value these two bytes is assigned to *len*. The timer2 is started with an elapse value of *len* plus two (*len* bytes of data plus two bytes CRC) times 10 times bit time and the method is left. If the conversion is not possible or *len* is smaller than 1, a message box is displayed, informing the user, that the data format was wrong. The variable *err* is set to true, *lock2* is set to false, *buff* and *inputData* are emptied, “0x00” is sent out via the serial port and the method is left.

If *text* contains less than two bytes, timer1 is started again with an elapse value of 10 times the bit time, which is the transfer time for one byte of data and the method is left. So it is waited for another byte to arrive.

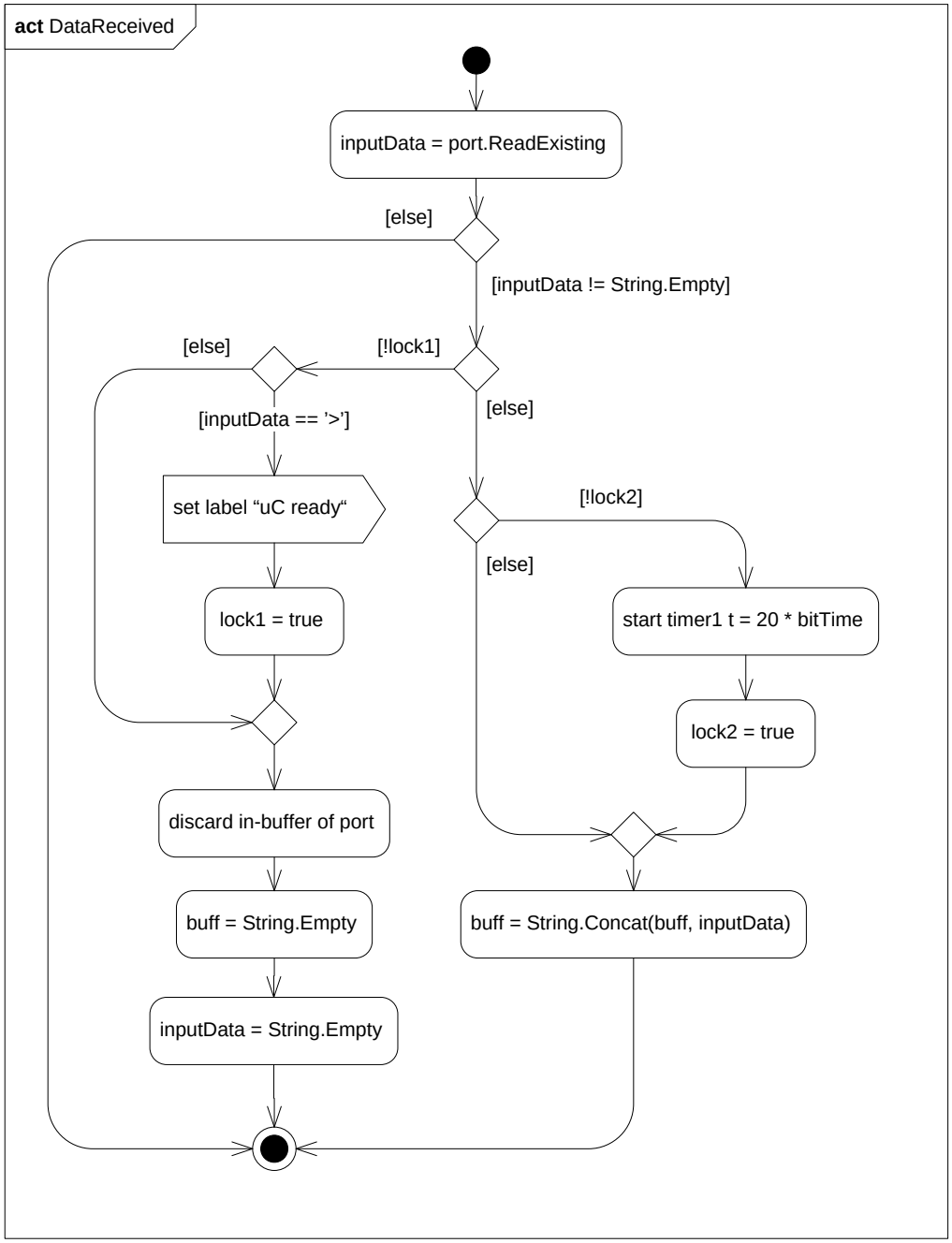


Figure 4.27: Activity diagram of DataReceived

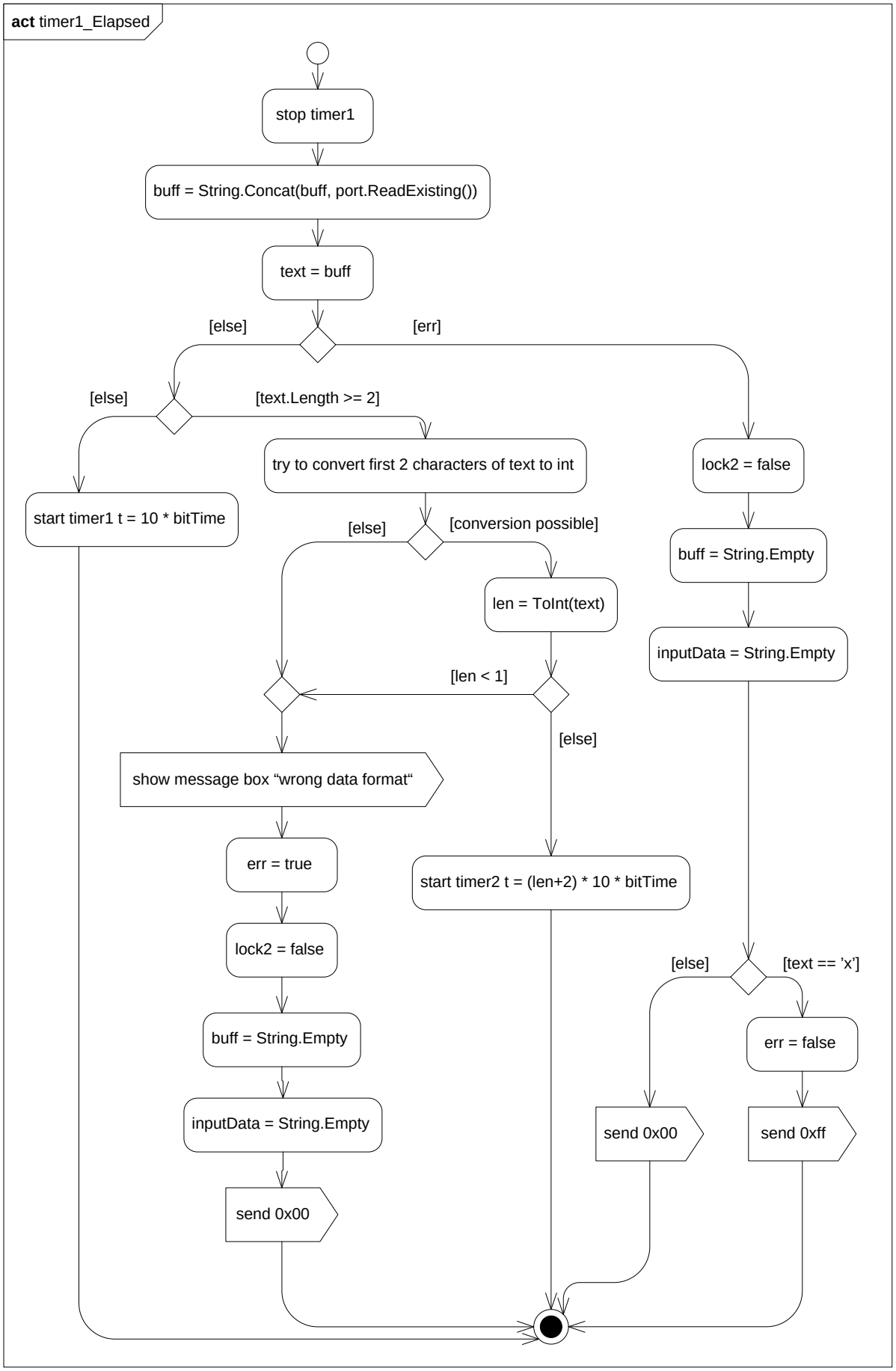


Figure 4.28: Activity diagram of timer1 elapsed

The activity diagram of the timer2 elapsed event is shown in Figure 4.29. Timer2 is stopped and the method ReadData is called.

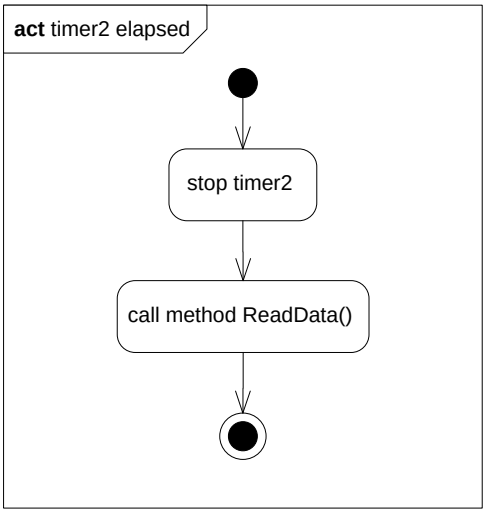


Figure 4.29: Activity diagram timer2 elapsed

The activity diagram of the method ReadData is shown in Figure 4.30. When the method is called, the contents of *buff* and the in-buffer of the serial port are merged. It is then checked if *buff* has a length bigger or equal to the length of data plus two bytes of CRC.

If *buff* has not the right length, timer2 is started again, with an elapse time value of 10 times the bit time (transfer time of 1 byte) and the method is left.

If *buff* has the required length, the transfer of the data package is complete and it can now be processed further. Therefore the content of *buff* is written into *received*, *buff* and *inputData* are emptied and the method ProcessInputD is called with the parameter *received*.

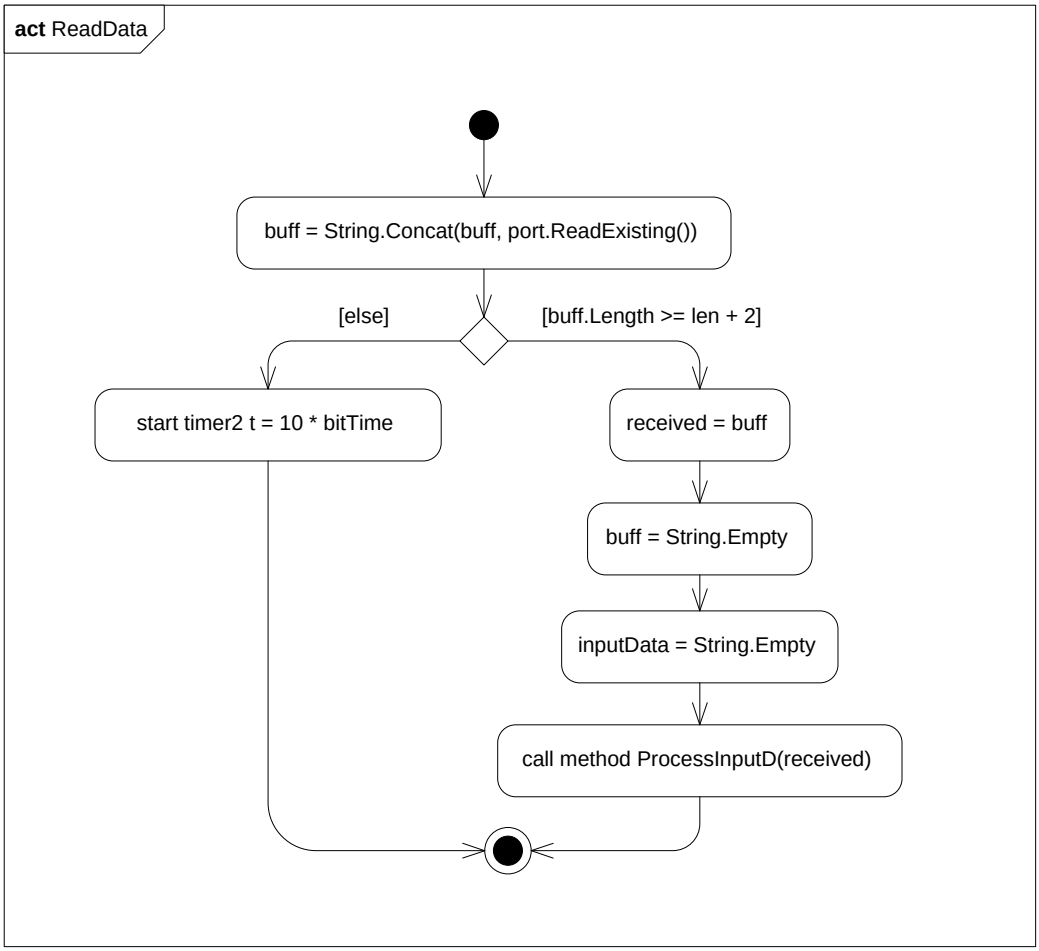


Figure 4.30: Activity diagram of method ReadData

The activity diagram of the method ProcessInputD is shown in Figure 4.31. When the method is entered the actual data is extracted from *received* and written into the variable *data*. The CRC of *data* is calculated and written into *crc*. The CRC calculation is done by the CRCTool written by de Wijs (27). Afterwards the CRC from *received* is extracted and written into *check*.

If the values of *crc* and *check* are not the same a transmission error occurred. Therefore *err* is set to true. The in-buffer of the serial port is discarded, “0x00” send sent out via the serial port and the method is left.

If the values of *crc* and *check* are the same, the transmission was successful and *inicomplete* is checked if the initialisation was already carried out.

If *inicomplete* is false, no initialisation has been carried out yet. Hence the variable *data* contains the number of data packages that will be transferred. It is tried to

convert *data* into a integer. If this conversion is not possible *err* is set to true and a message box is displayed that informs the user about the error. Afterwards the in-buffer of the serial port is discarded, “0x00” is sent out via the serial port and the method is left. If the conversion is possible *inicomplete* is set to true, the integer value of *data* is assigned to *pack* and the progress bar maximum is set to *pack*. The in-buffer of the serial port is discarded, “0xff” is sent out via the serial port and the method is left.

If *inicomplete* is true, *data* is written into the textbox, the progress bar value and *anz* are incremented and the in-buffer of the serial port is discarded. If *anz* is smaller than *pack*, “0xff” is sent out via the serial port and the method is left. Otherwise all data packages have been transferred. The status label is set to “transfer complete”, *anz* is set to zero and *incomplete* is set to false before the method is left.

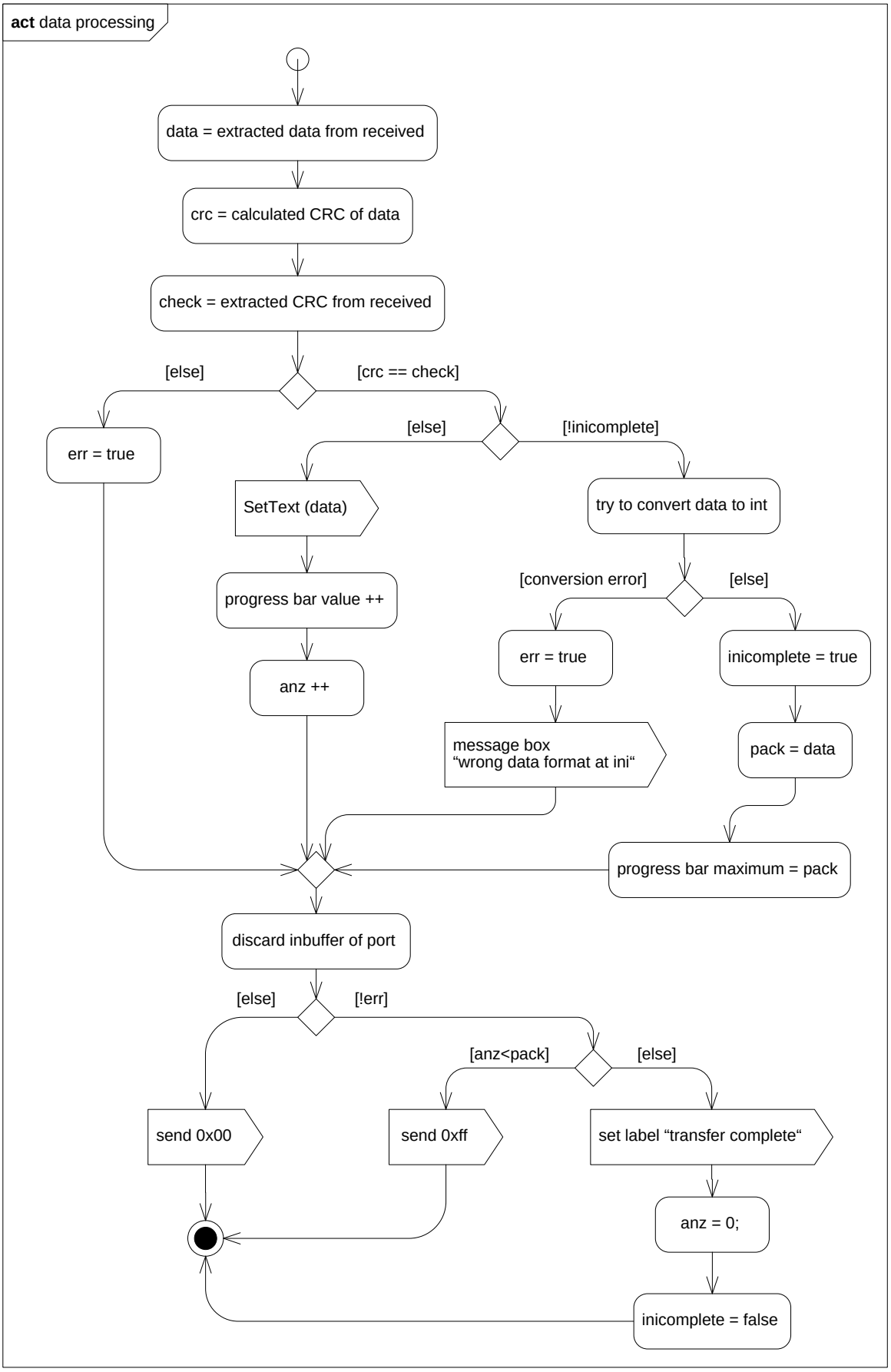


Figure 4.31: Activity diagram of data processing

5 Modelling of the HVAF Thermal Spraying System

For the purpose of closed loop control development, the HVAF thermal spraying system is modelled and simulated using Matlab/Simulink.

From the results of chapter 3 a model of the HVAF gun is built in section 5.1 “Modelling of the HVAF Gun”.

In section 5.2 “HVAF Thermal Spraying System Model” the model of the HVAF gun from section 5.1 is further developed to a simple model of the HVAF thermal spraying system.

The results of the simulation of the model generated in section 5.2 are presented in section 5.3 “Simulation Results”.

5.1 Modelling of the HVAF Gun

Based on chapter 3 “Theoretical Modelling of Combustion in the HVAF Gun” a model of the HVAF Gun is generated.

The modelling of the combustion chamber is described in section 5.1.1 “Combustion Chamber Model”:

A model of the accelerating nozzle is developed in 5.1.2 “Accelerating Nozzle Model”.

5.1.1 Combustion Chamber Model

The governing equations of a combustion process (see chapter 3.1) are a set of partial differential equations. To solve these equations, computational fluid dynamic (CFD) software is used. Coupling such software with a control algorithm like it is described by Candel (2) (see chapter 2.1) would go beyond the scope of this work. Thus in a first approach the adiabatic flame temperature (chapter 3.2.1) is utilised for the modelling of combustion in the combustion chamber.

Using Equation (3.9) the model for the adiabatic flame temperature is built (Figure 5.1). Additionally the calculation of the ratio of specific heats and the specific gas constant of the exhaust gas is performed in this block. While it is assumed that the ratio of specific heats of the exhaust gas is constant, the calculation of the specific gas constant of the exhaust gas is done by (20):

$$R_s = \frac{R}{M} \tag{5.1}$$

Where R_s is the specific gas constant of the exhaust gas, R is the ideal gas constant and M is the molar mass of the exhaust gas.

The time delays for specific gas constant and for the temperature are inserted mainly to avoid an algebraic loop, which is described more detailed in section 5.2, but also in reality the chemical reactions are not infinitely fast.

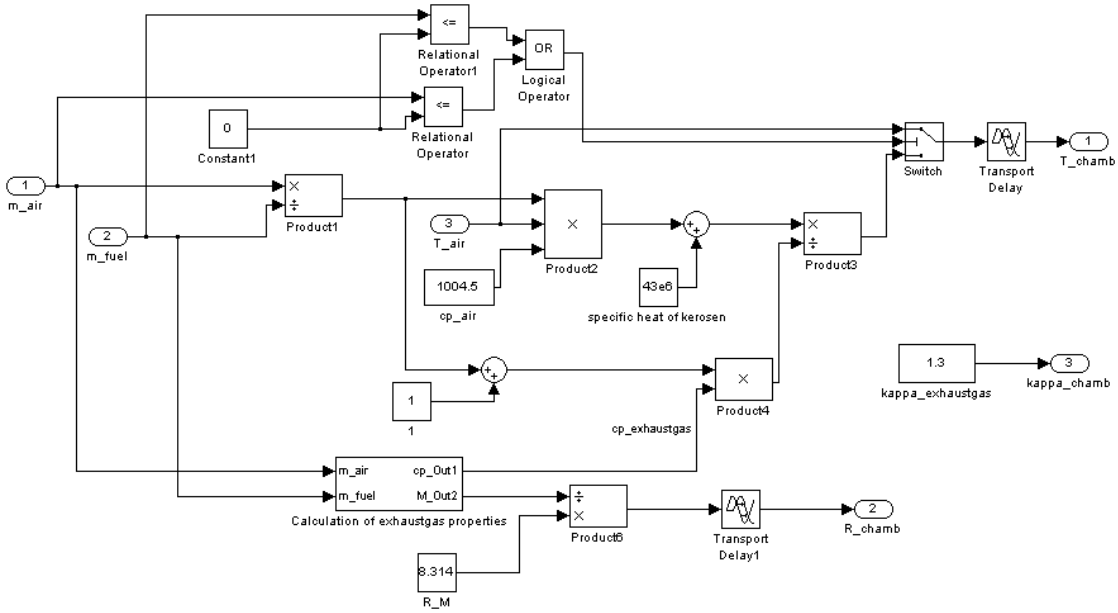


Figure 5.1: Simulink model of the adiabatic flame temperature

For calculating the properties of the exhaust gas, the assumptions of chapter 3.2.2 are adopted. Furthermore it is assumed, that the specific heats of the exhaust gas components are temperature-independent. The properties of the exhaust gas are calculated according to equations (3.11) and (3.12) and the corresponding model is shown in Figure 5.2.

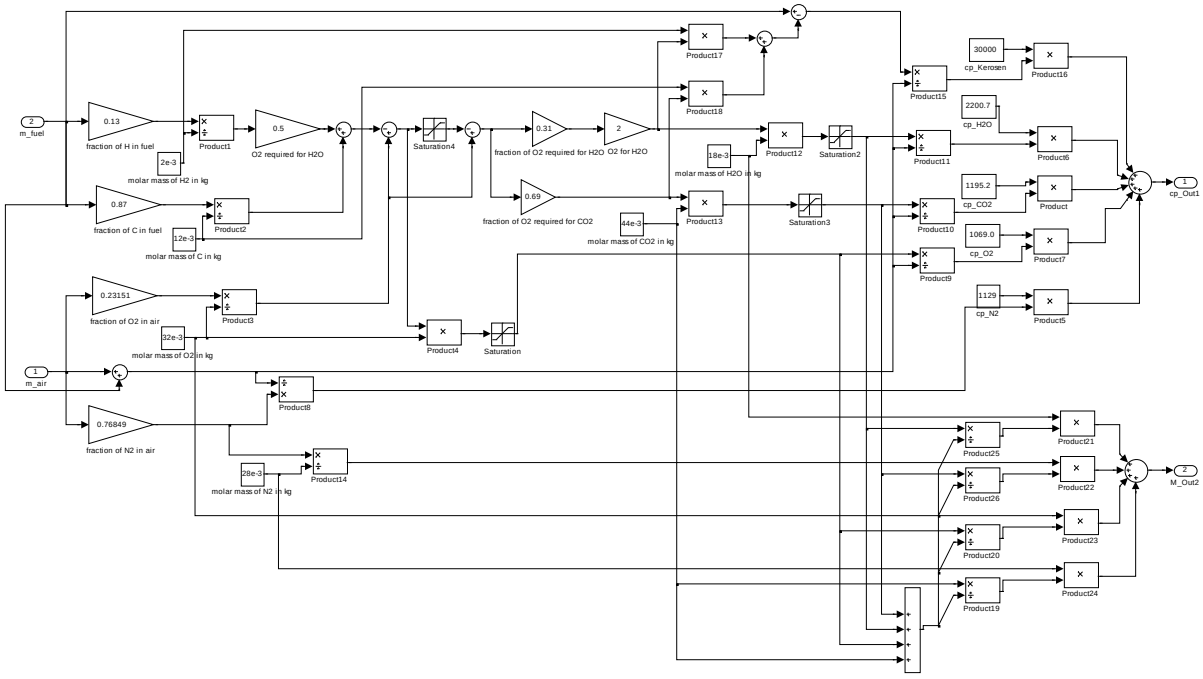


Figure 5.2: Simulink model of the calculation of the exhaust gas properties

It is assumed that pressure and temperature in the combustion chamber are constant and location-independent. Hence the model of the combustion chamber is zero dimensional. Furthermore it is assumed that the ideal gas law is applicable. In this case the pressure in the combustion chamber can be calculated by (20):

$$p_{chamber} = \frac{m\, R_{exhaust}\, T_{exhaust}}{V_{chamber}} \tag{5.2}$$

Where $p_{chamber}$ is the pressure in the combustion chamber, m is the mass in the combustion chamber, $V_{chamber}$ is the volume of the combustion chamber, $T_{exhaust}$and $R_{exhaust}$ are temperature and specific gas constant of the exhaust gas.

Due to the fact, that a zero dimensional, adiabatic combustion chamber is assumed, the state of the exhaust gas is the state of the gas in the chamber.

The model of the combustion chamber (Figure 5.3) is generated by combining the adiabatic flame temperature with the ideal gas law. The mass in the combustion chamber is calculated by integrating the sum of the incoming and outgoing mass flows.

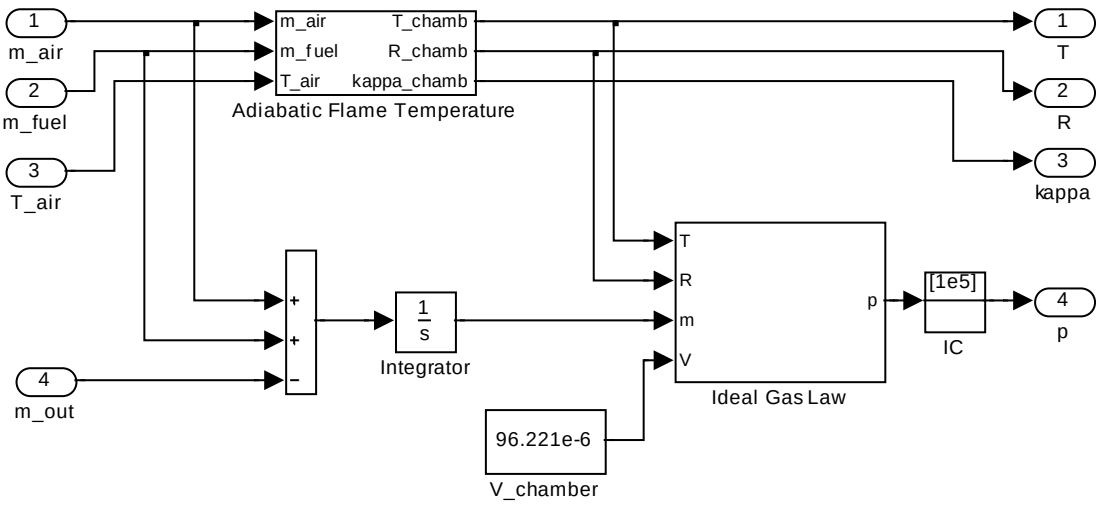


Figure 5.3: Simulink model of the combustion chamber

5.1.2 Accelerating Nozzle Model

A detailed description of the accelerating nozzle is given in chapter 3.3. Therefore the Simulink model of the accelerating nozzle is generated from the Equations of chapter 3.3 . Equation (3.15) leads to the model for the velocity generated by the accelerating nozzle, shown in Figure 5.4.

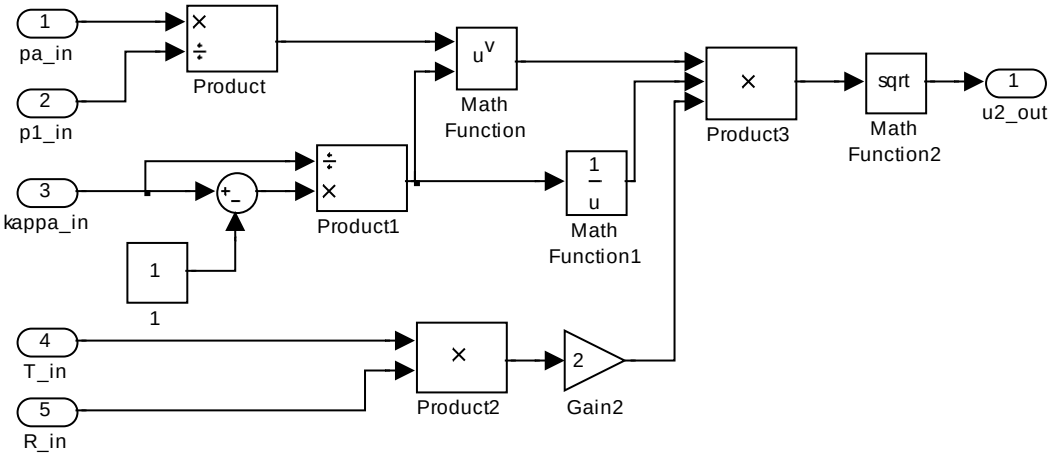


Figure 5.4: Simulink model velocity generated by the accelerating nozzle

From equation (3.16) the model for the mass flow rate through the nozzle is derived (Figure 5.5).

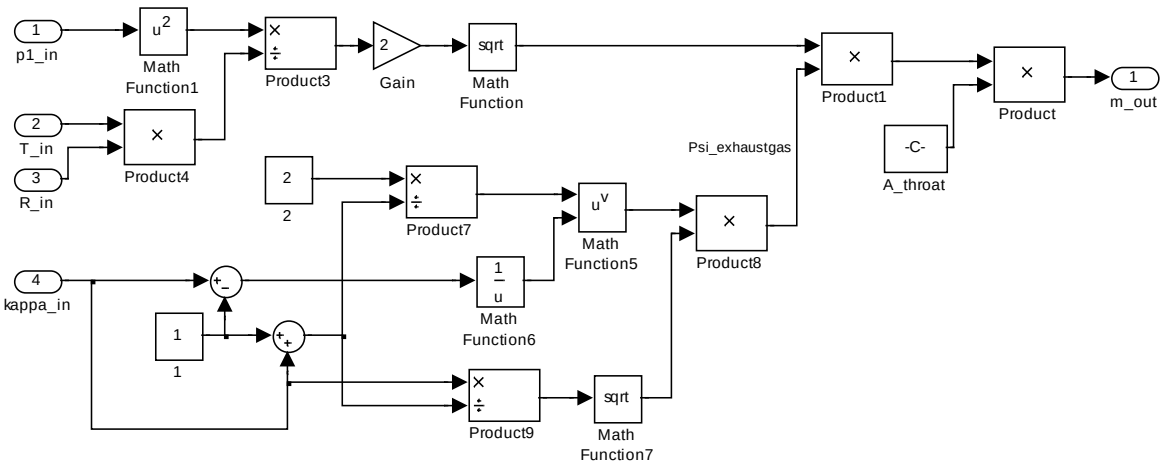


Figure 5.5: Simulink model of the mass flow rate through the accelerating nozzle

Merging of the velocity and the mass flow rate model leads to the model of the accelerating nozzle shown in Figure 5.6.

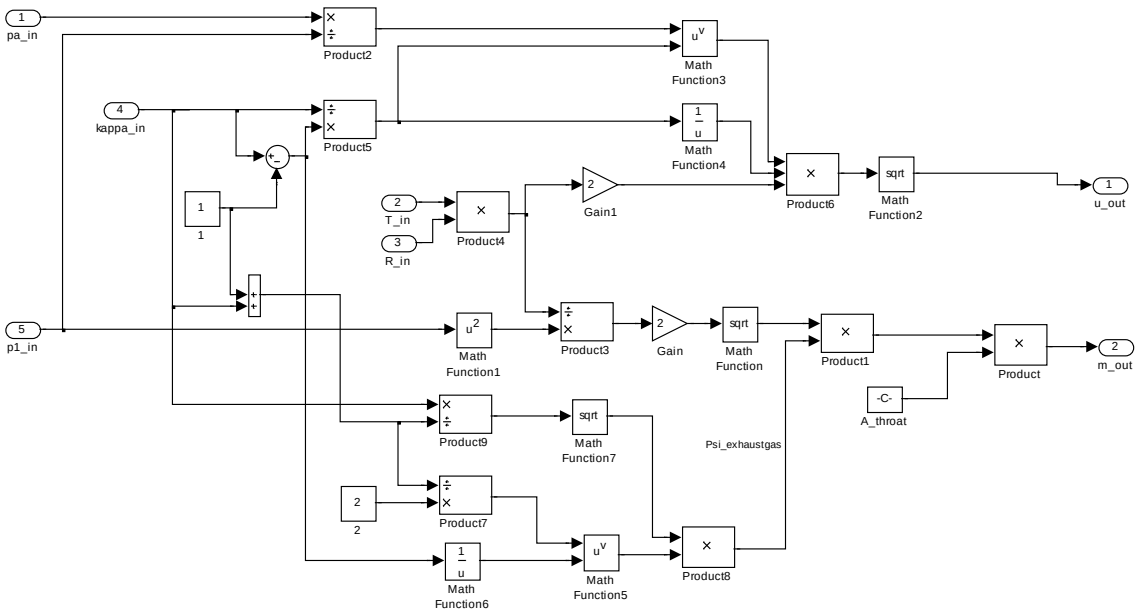


Figure 5.6: Simulink model of the accelerating nozzle

5.2 HVAF Thermal Spraying System Model

By combining the models of the combustion chamber and the nozzle, the model of the HVAF gun is generated (Figure 5.7). For simulation, this model needs to be expanded by a fuel and an air supply. Hence a model of the HVAF thermal spraying system is required.

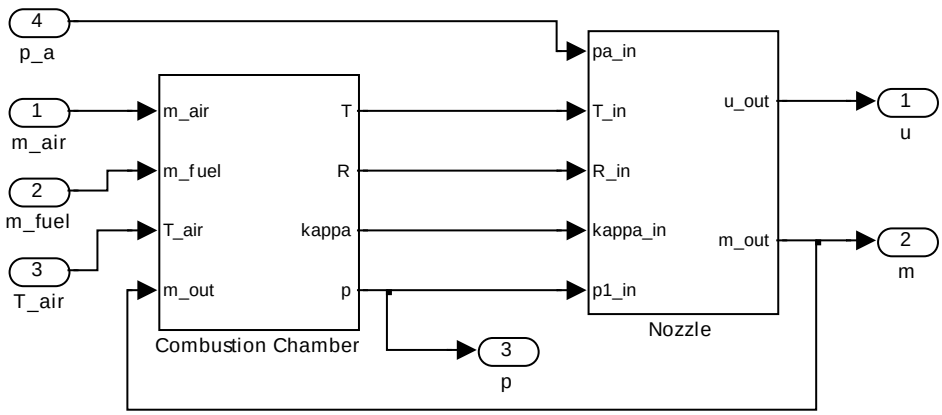


Figure 5.7: Simulink model of the HVAF gun

For modelling the fuel system the simple approach of a pipe with a pressure drop due to friction and turbulence is used (Figure 5.8).

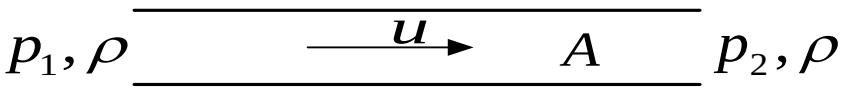


Figure 5.8: Pipe representing the fuel system

Using the Bernoulli equation, neglecting the influence of gravity and expanding it by the loss of energy by dissipation (20):

$$\frac{p_1}{\rho} = \frac{p_2}{\rho} + Y_d \tag{5.3}$$

Where p_1 is the pressure at the inlet of the pipe, p_2 is the pressure at the outlet of the pipe, ρ is the density of the fuel and Y_d is the loss of energy by dissipation.

The loss of energy by dissipation can be calculated by (20):

$$Y_d = \alpha^{-2} \frac{u^2}{2} \tag{5.4}$$

Where u is the flow velocity in the pipe and α is the flow coefficient, taking into account friction and turbulence.

Combining equations (5.3) and (5.4) the flow rate is:

$$\dot{V} = A\,u = \alpha\,A \left[\frac{2\,(p_1 - p_2)}{\rho}\right]^{\frac{1}{2}} \tag{5.5}$$

Where A is the cross-sectional area of the pipe.

The model of the fuel system (Figure 5.9) can now be derived from equation (5.5) using the pressure of fuel p_{fuel} instead of p_1 and the combustion chamber pressure $p_{chamber}$ instead of p_2. To represent the proportional valve, the cross-sectional area can be modified by multiplying it with a factor.

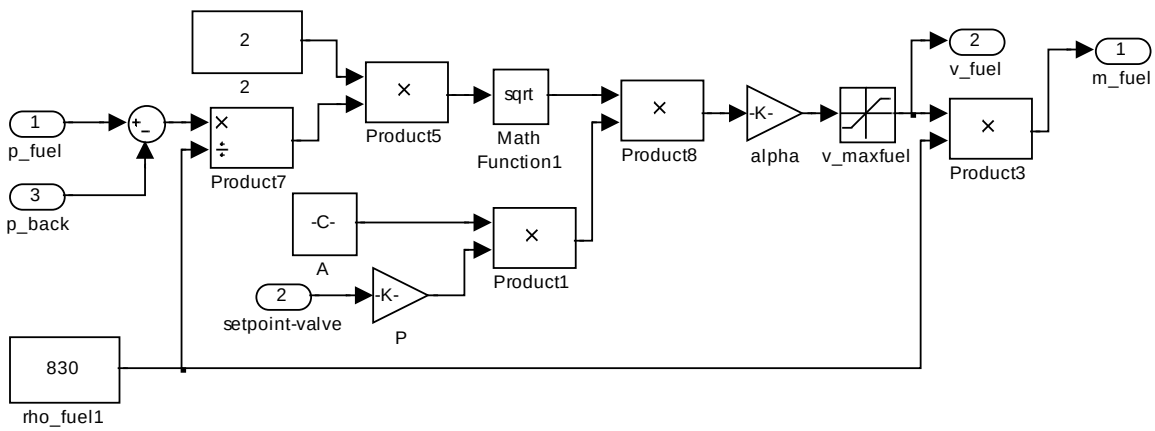


Figure 5.9: Simulink model of the fuel system

Similar to the fuel system, the air system can be modelled. According to Sigloch (20) gas flows with a Mach number $Ma < 0.3$ can be treated as incompressible, making an error $err < 5\%$. It is therefore assumed that the air flow is incompressible. The

density of air is calculated with equation (4.2) (see section 4.2.2). Figure 5.8 shows the model of the air system.

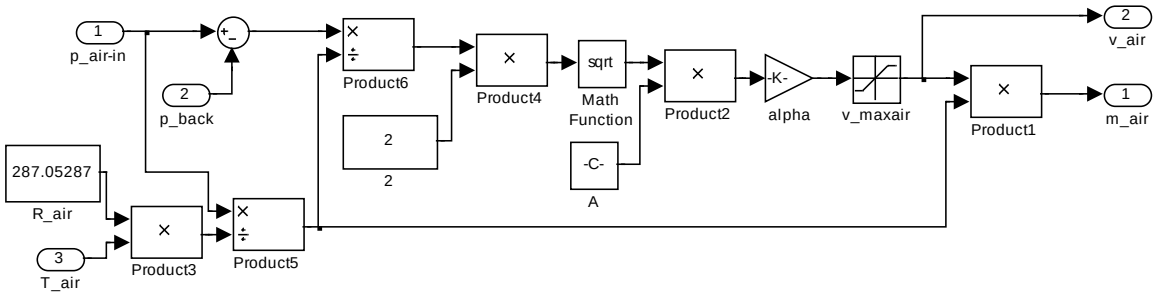


Figure 5.10: Simulink model of the air system

The combination of the models of the combustion chamber model, the air system and the fuel system lead to the model of the HVAF thermal spraying system shown in Figure 5.10.

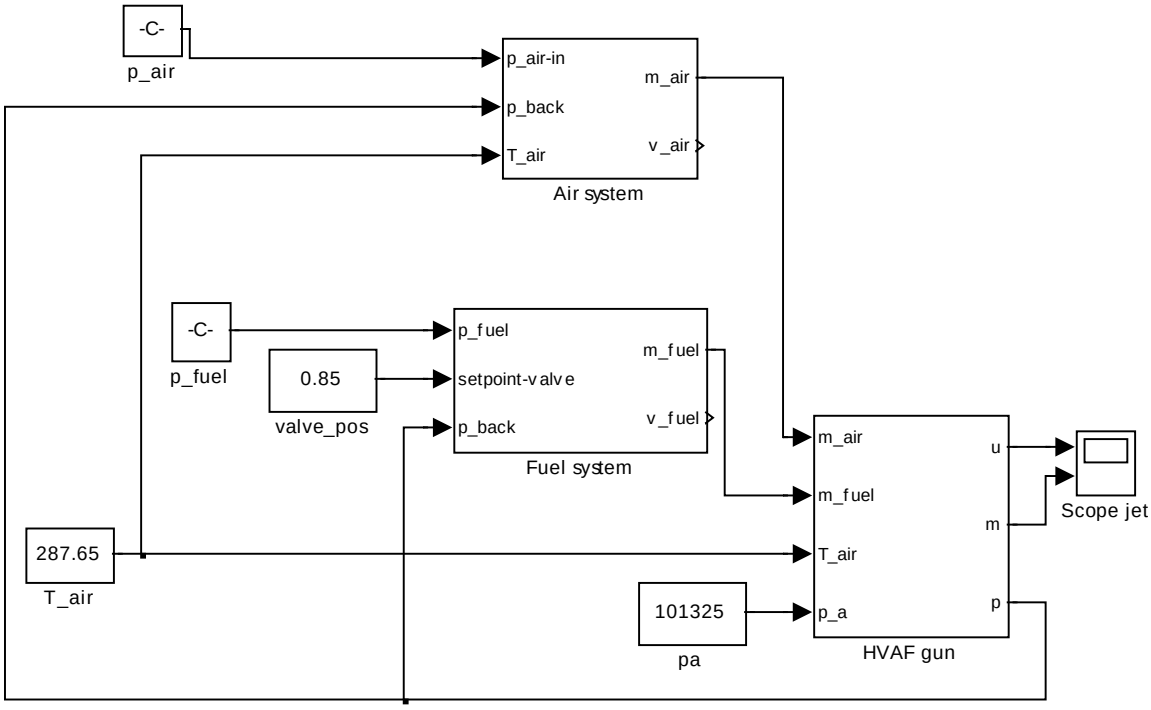


Figure 5.11: Simulink model of the HVAF thermal spraying system

As mentioned in section 5.1.1, there are delay times in the adiabatic flame temperature model, to prevent an algebraic loop.

An algebraic loop occurs if the forward and the feedback branch of a signal path only consist of direct feedthrough blocks. Direct feedthrough blocks, are blocks where the input signals are directly passed to the output, such as *Gain*, *Product* or *Sum* blocks. So if an algebraic loop is present, in one calculation step the input signal is passed to the output with no time delay and this output signal is passed to the input with no time delay. Meaning input signal is output signal is input signal! Or action is reaction is action! Therefore algebraic loops should be avoided. (28)

If there would be no time delays in the adiabatic flame temperature, there would be an algebraic loop in the model. As the temperature in the combustion chamber depends on the mass flow rates of fuel and air, the pressure in the combustion chamber depends on the temperature in the combustion chamber and the mass flow rates of fuel and air depend on the pressure in the combustion chamber.

5.3 Simulation Results

For evaluation of the above built model, selected signals from the model are compared with measurement data of the real process. The measurement data was obtained with the control system described in chapter 4.

Prior to this evaluation the model was parameterised by adjusting the parameters A and α in the air and the fuel system blocks according to measurements that where performed in a narrow range around the typical operating point of the HVAF gun.

As the control of combustion in the HVAF thermal spraying process is an operating point control, oscillations in the signals are of minor interest, as long as the amplitude is small and the mean value can be seen as constant.

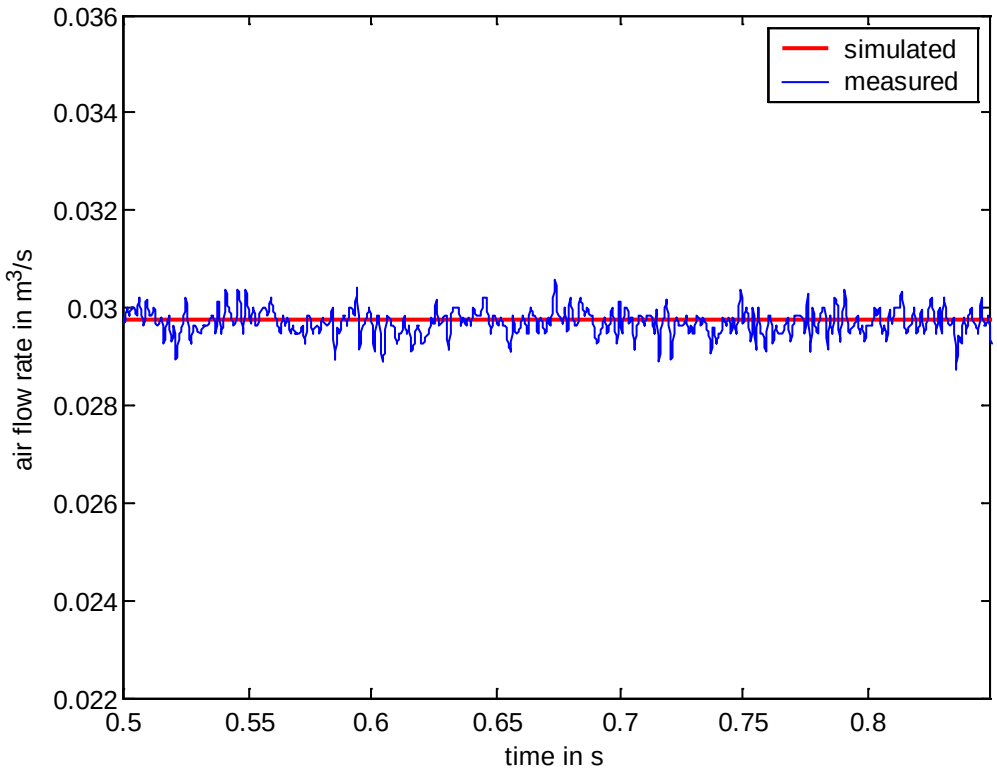


Figure 5.12: Comparison of measured and simulated air flow rate

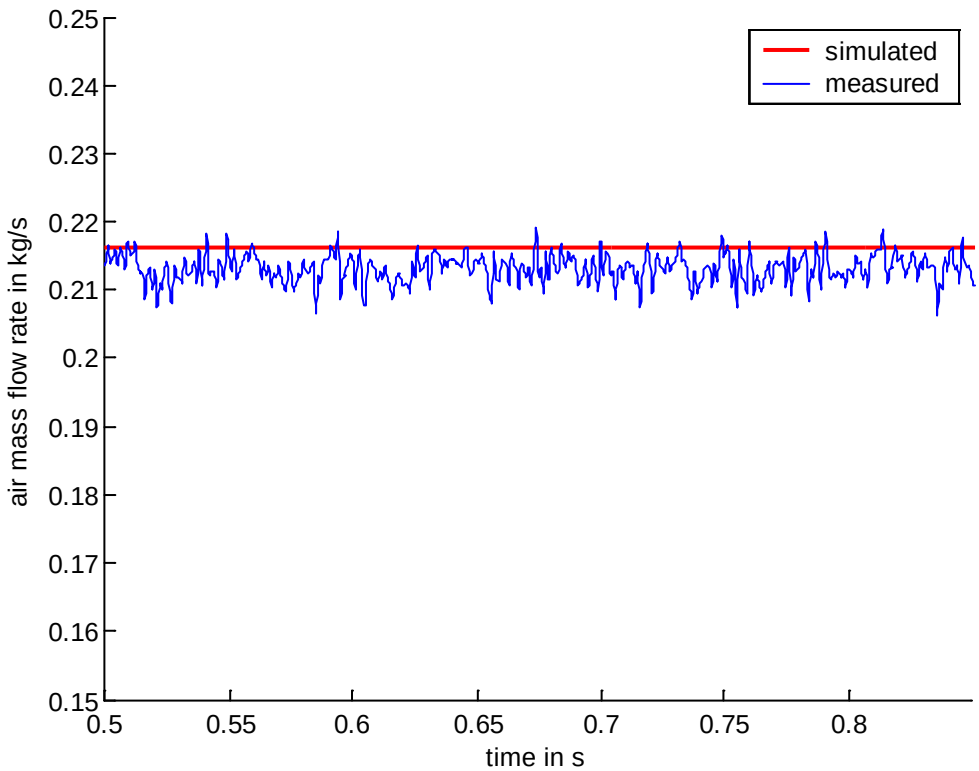


Figure 5.13: Comparison of measured and simulated air mass flow rate

In Figure 5.11 the simulated and the measured air flow rate are compared. As described in chapter 4 the mass flow rate of air cannot be measured directly by control system for the HVAF thermal spraying process. Thus it must be calculated by equation (4.2) from the measurement data of the air pressure, the air flow rate and the air temperature sensor. This results in an adding up of the measurement errors. Due to this fact the focus of the parameterisation of the model was on the flow rate and not on the mass flow rate of air. This is why in Figure 5.13 the measured value of air mass flow rate is slightly different from the calculated value.

The comparison of the calculated and measured fuel flow rate is shown in Figure 5.13. An additional comparison of calculated and measured fuel mass flow rate is not necessary, as it would be the same diagram, just multiplied by the density of fuel.

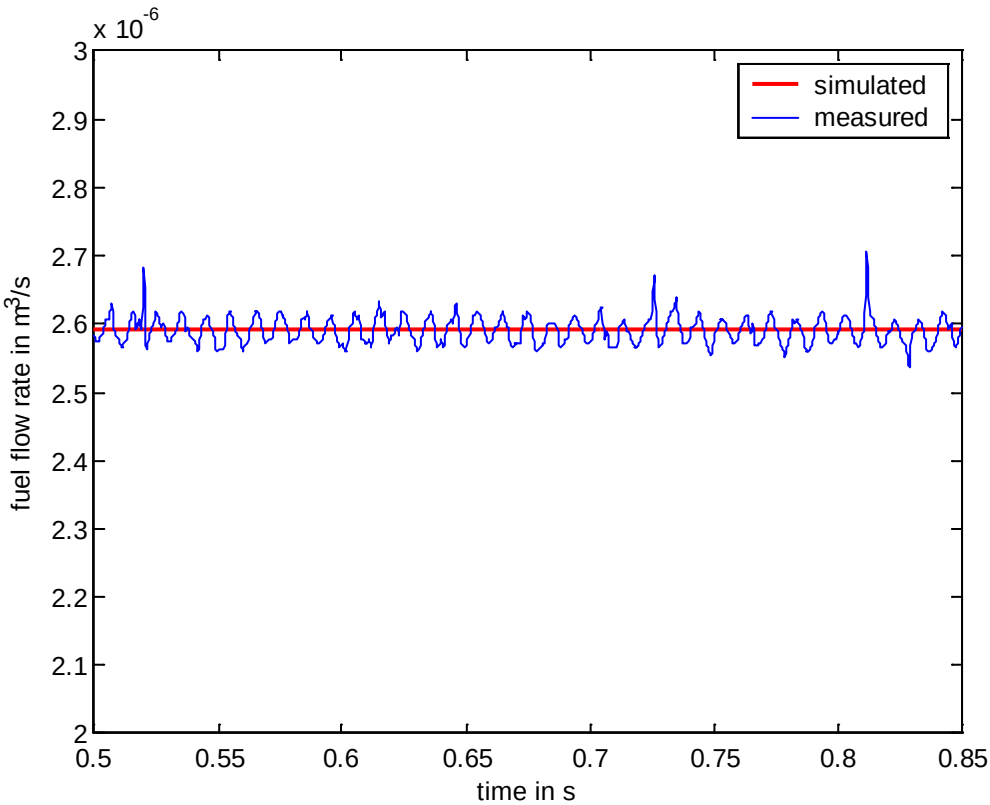


Figure 5.14: Comparison of measured and simulated fuel flow rate

It can be seen from the above showed diagrams that the model corresponds well with the measurement data, although it was built with many simplifying assumptions. This is due to the fact that the parameterisation of the model was performed with a set of data that has a narrow range of variation. So this model represents the HVAF thermal spraying system only in proximate area of the typical operating point.

6 Conclusion

In this research, a controller, based on a Microchip PIC24H microcontroller, for the HVAF thermal spray system was developed. The significance of this research is that the developed controller is a completely novel design, as there are no prototypes of HVAF controllers available. Controlling of a highly unstable, multivariable combustion process of the HVAF spray gun is a complex task. The controller provided reliable control of the system within the prescribed range of combustion parameters, including the auxiliary equipment. The controller allows collecting and storing the process parameters, which are important for quality control in thermal spraying. It is envisaged that the controller can be interfaced with higher level feedback systems (lasers and optical) for on-line monitoring of the direct thermal spraying parameters, such as a coating thickness and a temperature in real time, and make necessary corrections of the thermal spray gun parameters.

The use of a PIC24H microcontroller in an industrial environment proved to be complicated as a low voltage (3.3V) microcontroller an adequate interface to the standard voltage level (24V) in control panels had to be implemented. The fact that it is a low cost microcontroller with a free development environment (MPLAB), and free C-compiler (C30) cannot be used as main determining factors when selecting a microcontroller. For example, it required considerable time to fix incomplete header files and linker scripts. Despite these problems a fully operational PIC24H based controller was developed.

The Controller is able to start and run the HVAF thermal spray process in a manual mode, where the flow rate of fuel is set by the operator as well as in an automatic mode, where the flow rate of fuel is set by the controller. Furthermore all relevant process data can be monitored, captured and stored. This builds the base for further research work.

The model of the HVAF thermal spraying system derived in this work is basic, but in the tight operating range the model was designed for, it corresponds well with measurement data of the real process.

Possible future developments are:

Perform a more detailed modelling of the combustion process, towards an advanced level of control. If the combustion can be controlled precisely, the next step to an

overall process would be the modelling of the modelling of particle behaviour. This could lead to an advanced model based control similar to that used in HVOF thermal spraying.

Investigate if the flame glow plug can be used as sensor. Since the resistance of the heating coil of the flame glow plug is depending on temperature, it is the question if the required accuracy can be achieved.

Determine the maximum switching frequency of the 2/3 valve for ignition fuel and possibly add a MOSFET based digital output, to perform experiments on pulsed secondary fuel injection. Investigate the effects on operating range, stability and coating results/quality.

On the control system a LCD screen or similar could be implemented to realise graphical users interface. Furthermore a USB interface could be implemented instead of RS232.

References

1. Annamalai K, Puri IK. Combustion Science and Engineering. 1st ed. Boca Raton (FL): CRC Press; 2007.

2. Candel S. Combustion Dynamics and Control: Progress and Challenges. Proceedings of the Combustion Institute. 2002; 29: p. 1-28.

3. Davis JR, editor. Handbook of Thermal Spray Technology. 2nd ed. Materials Park (OH): ASM International; 2005.

4. Pawlowski L. The Science and Engineering of Thermal Spray Coatings. 1st ed. Chichester: John Wiley & Sons; 1995.

5. Gorlach IA. Thermal Stress Evalutation of Thermo-Blast Jet Nozzle Materials. PhD-thesis. Potchefstroom:; 2004.

6. Gorlach IA. A new method for thermal spraying of Zn-Al coatings, Thin Solid Films. International Journal of the Science and Technology of Condensed Matter Films. 2009; 517(17): p. 5270-5273.

7. Gorlach IA. Low-cost HVAF system for thermal spraying of WC-Co. Proceedings of the International Thermal Spray Conference (ITCS). 2009;: p. 718-721.

8. Docquire N, Candel S. Combustion control and sensors: a review. Progress in Energy and Combustion Science. 2002;(28): p. 107-150.

9. Franklin GF, Powell JD, Emami-Naeini A. Feedback Control of Dynamic Systems Upper Saddle River (NJ): Pearson Education; 2006.

10. Yu KH, Wilson KJ, Schadow KC. Liquid-Fueled Active Instability Suppression. Twenty-Seventh Symposium on Combustion/The Combustion Institute. 1998;: p. 2039-2046.

11. Hermann J, Orthmann A, Hoffmann S, Berenbrink P. Combination of Active Instability Control and Passive Measures to Prevent Combustion Instabilities in a 260MW Heavy Duty Gas Turbine. NATO RTO Meeting on Active Control Technology. 2000.

12. Kalogirou SA. Artificial intelligence for the modeling and control of combustion processes: a review. Progress in Energy and Combustion Science. 2003;(29): p. 515-566.

13. Blonbou R, Laverdant A, Zaleski S, Kuentzmann P. Active Control of Combustion Instabilities on a Rijke Tube using Neural Networks. Proceedings of the Combustion Institute. 2000;(28): p. 747-755.

14. Li M, Christofides PD. Modeling and Control of High-Velocity Oxygen-Fuel (HVOF) Thermal Spray: A Tutorial Review. Journal of Thermal Spray Technology. 2009.

15. Li M, Shi D, Chirstofides PD. Diamond Jet Hybrid HVOF Thermal Spray: Gas-Phase and Particle Behavior Modeling and Feedback Control Design. Industrial & Engineering Chemistry Research. 2004;(43): p. 3632-3652.

16. Gerlinger P. Numerische Verbrennungssimulation. 1st ed. Berlin Heidelberg: Springer-Verlag; 2005.

17. Epple B, Leithner R, Linzer W, Walter H. Simulation von Kraftwerken und wärmetechnischen Anlagen. 1st ed. Epple B, Leithner R, Linzer W, Walter H, editors. Wien, New York: Springer; 2009.

18. Warnatz J, Maas U, Dibble RW. Combustion. 2nd ed. Berlin, Heidelberg: Springer; 1999.

19. Glück B. Zustands- und Stoffwerte, Verbrennungsrechnung. 2nd ed. Berlin: Verlag für Bauwesen; 1991.

20. Sigloch H. Technische Fluidmechanik. 6th ed. Berlin Heidelberg: Springer-Verlag; 2008.

21. Weilkiens T. Systems Engineering with SysML/UML. 1st ed. Amsterdam: Elsevier / Morgan Kaufmann; 2007.

22. Object Management Group Inc. OMG System Modeling Language. [Online].; 2010 [cited 2010 January 13. Available from: http://www.omgsysml.org/.

23. Saal O. RS232-Schnittstelle. [Online].; 2007 [cited 2010 January 18. Available from: http://www.oliver-saal.de/elektronik/rs232.php.

24. Microchip Technology Inc. PIC24H Reference Manual. 2009; Revision B.

25. Barr M, Massa A. Programming Embedded Systems. 2nd ed. Beijing: O'Reilly; 2006.

26. Microsoft Corporation. Microsoft Developer Network (MSDN) Library. 2009.

27. De Wijs M. CRC Encoding. [Online].; 2003 [cited 2010 January 18. Available from: http://www.codeproject.com/KB/cs/marcelcrcencoding.aspx.

28. Bode H. Matlab-Simulink, Analyse und Simulation dynamischer Systeme. 2nd ed. Wiesbaden: Teubner; 2006.

Appendix

Appendix A

Program listings (on attached CD-ROM)

Appendix B

Matlab/Simulink models (on attached CD-ROM)

Appendix C

Data sheets (on attached CD-ROM)

Appendix D

Operation Manual for the Control System for the HVAF Thermal Spray Process

The operation of the control system for the HVAF thermal spray process is described in the following manual.

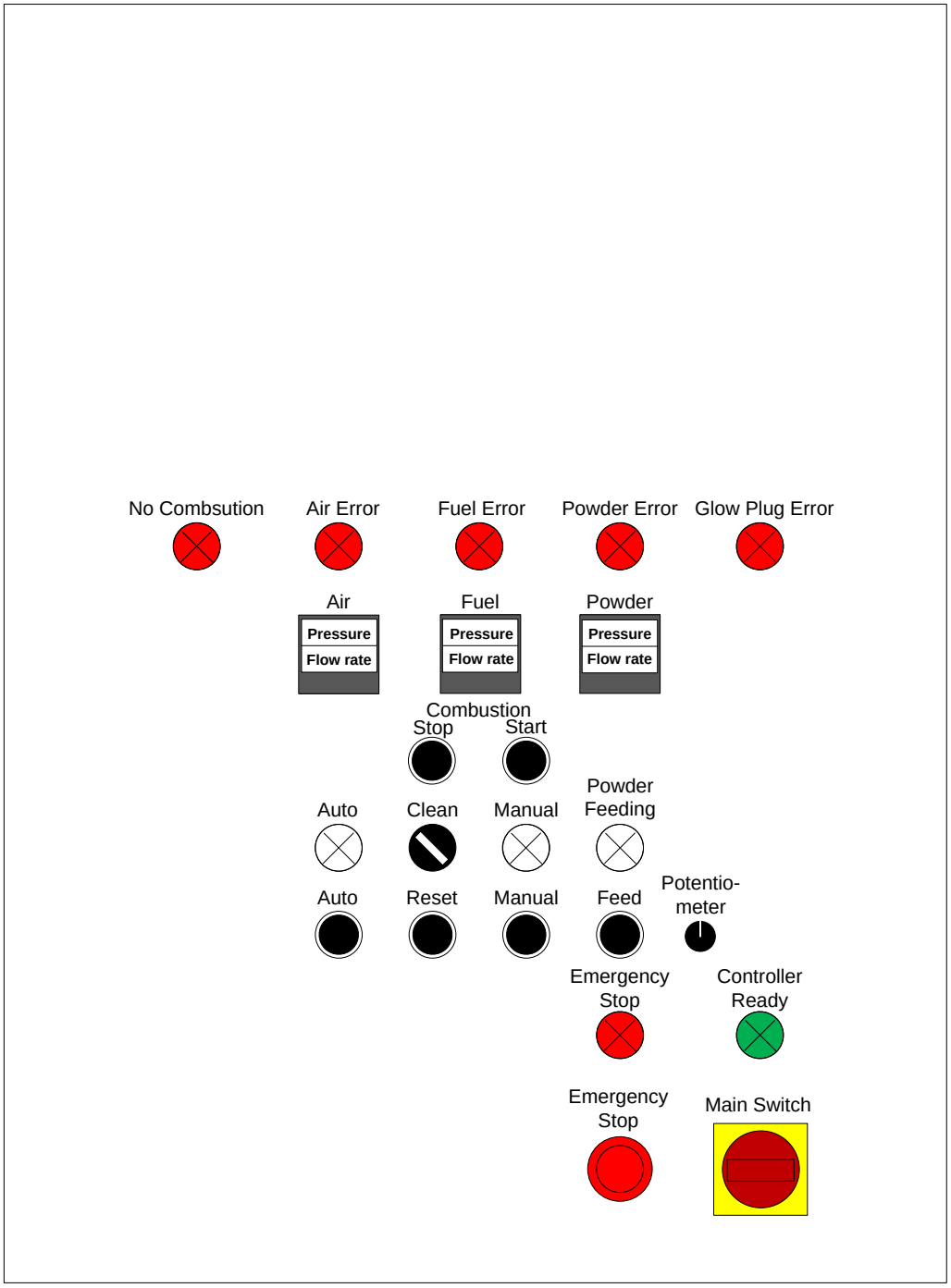


Fig. 1: User interface of the control system

The control system features four different operation modes, which are described in this manual. These operation modes are: “Emergency Stop”, “Clean”, “Manual Mode”, “Data Transfer” and “Automatic Mode”.

The user interface of the control system is shown in Fig. 1.

To use the control system make sure that all necessary air and fuel lines are connected to the HVAF gun, the powder feeder and to the control system. Make sure that the air lines are pressurised and that enough fuel is in the fuel tank. Check the connection between the glow plug and the control system. Plug the mains plug of the control system in and switch on the Main switch.

If the Controller Ready light is not switched on, there’s either no voltage supply to the control system or the microcontroller in the control system is not running correctly.

If the Controller Ready light is switched on, the control system is ready for operation and can be used as described in the following sections.

Emergency Stop

To initiate an Emergency Stop, push the Emergency Stop button. The Emergency Stop light is switched on.

If the reason for the Emergency Stop is not present anymore, unlock the Emergency Stop button and press Reset. The Emergency Stop light will be switched off and the Controller Ready light will be switched on. The control system is now ready for normal operation, again.

Clean

To perform a simple cleaning of the HVAF gun, switch the Clean switch on. The main air supply to the HVAF gun is switched on as long as the Clean switch is on.

Manual Mode

To operate in manual mode, push the Manual button. The Manual light will be switched on as soon as the control system is in manual mode.

By pushing the Feed button, the powder feeder can be tested and adjusted. The Powder Feeding light is switched on as long as the powder feed signal is given to the powder feeder. If the Feed button is pushed again, the powder feed signal is switched of, the Powder Feeding light is switched off and the normal operation can be carried on.

To start the combustion, press Combustion Start. The fuel flow can be adjusted by turning the Potentiometer. If the combustion has stabilised at the desired operating point, the spraying can be started by pushing the Feed button. The Powder Feeding light is switched on to signal that now powder is feed to the HVAF gun. To stop spraying, press the Feed button again. The powder feeding to the HVAF gun will be stopped and the Powder Feeding light will be switched off. To stop the combustion, press the Combustion Stop button. After ten seconds of cooling the HVAF gun, the control system is ready for further operation.

During the whole operation, process data can be captured by pushing the Auto button.

The manual mode can be left, by pushing the Reset button.

Data Transfer

To transfer captured process data to an external PC, the following procedure must be executed.

Start the PC application “HVAF data”, the graphical user interface shown in Fig. 2 will appear. Select the serial port to which the control system for the HVAF thermal spray process is connected. Push the Feed button on the user interface of the control system. If a connection between control system and PC has been established, the label on the graphical user interface of the PC application changes to “uC ready”. Click now on the button Connect on the graphical user interface of the PC application. The received data will appear in the textbox of the graphical user interface. If the transfer of the process data is complete the label will change to

“transfer complete”. The process data can now be saved, by clicking on the button Save.

To get the control system ready for normal operation again, push the Reset button.

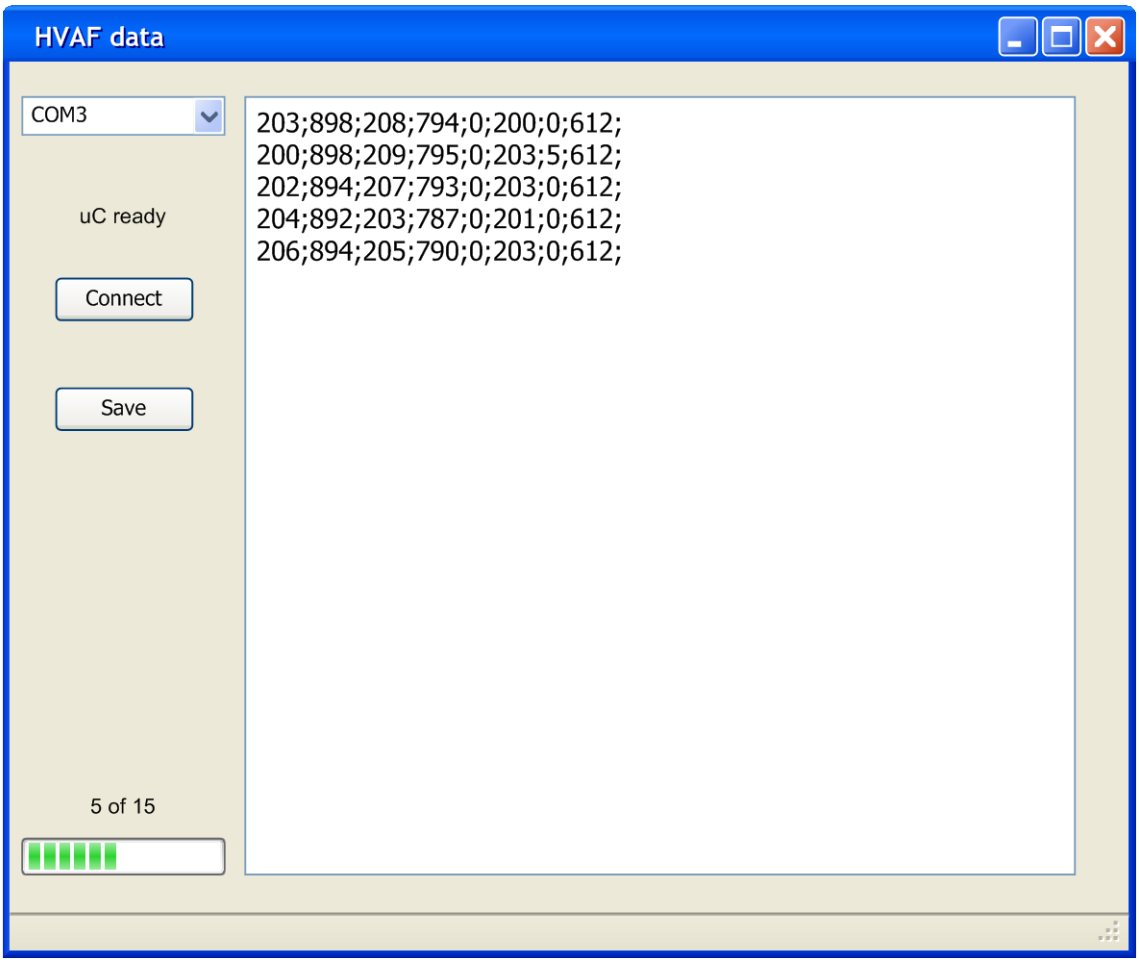


Fig. 2: Graphical user interface of the PC application

Automatic Mode

To operate in automatic mode, push the Auto button. It is tested if the main air supply is sufficient for operation of the HVAF gun. This is done by switching on the main air for five seconds and checking the resulting air pressure.

If the pressure of the main air is below 5.5 bar, the air supply is not sufficient. Therefore the automatic mode is left and the Air Error light is switched on. By pushing the Reset button normal operation can be carried on.

If the pressure of the main air is above 5.5 bar, the air supply is sufficient. The Auto light is switched on and the control system is ready to operate in automatic mode.

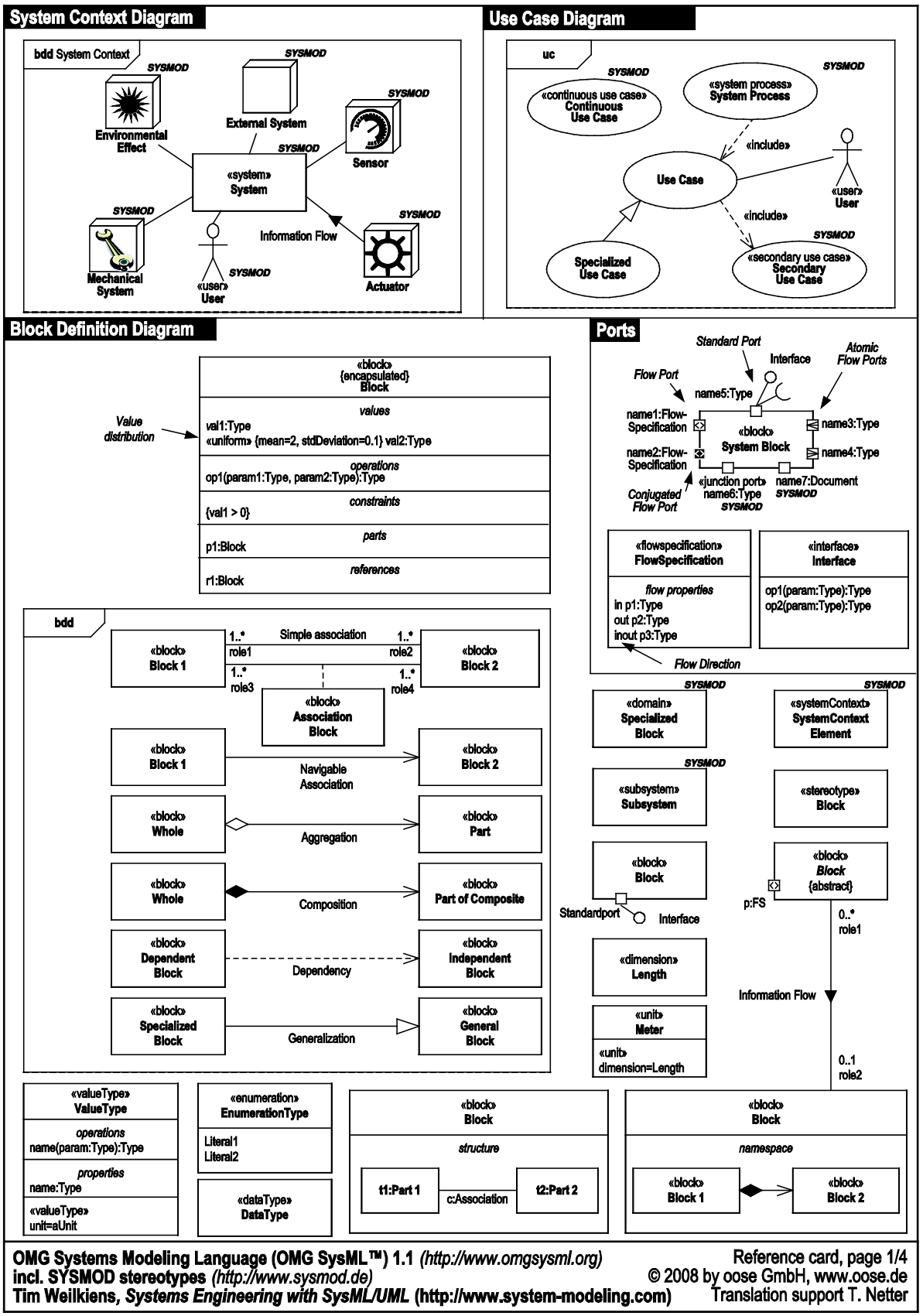
To start the combustion, press Combustion Start. By pressing the Combustion Start button, the fuel/air ratio can be increased. If the combustion has stabilised at the desired operating point, the spraying can be started by pushing the Feed button. The Powder Feeding light is switched on to signal that now powder is feed to the HVAF gun. To stop spraying, press the Feed button again. The powder feeding to the HVAF gun will be stopped and the Powder Feeding light will be switched off. To stop the combustion, press the Combustion Stop button. After ten seconds of cooling the HVAF gun, the control system is ready for further operation.

During the whole operation, the pressure of the air supply for the powder feeder is monitored. If this pressure drops below 6.5 bar, normal operation is aborted and the Powder Error light is switched on. By pushing the Reset button normal operation can be carried on.

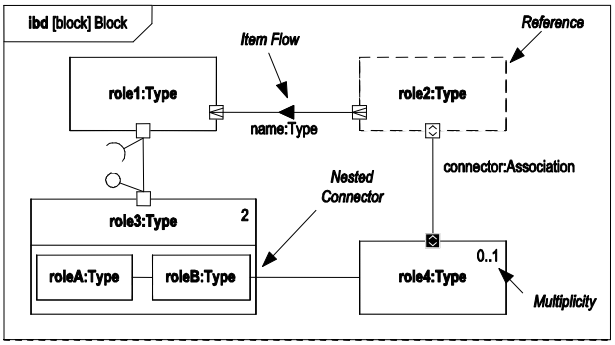
The automatic mode can be left, by pushing the Reset button.

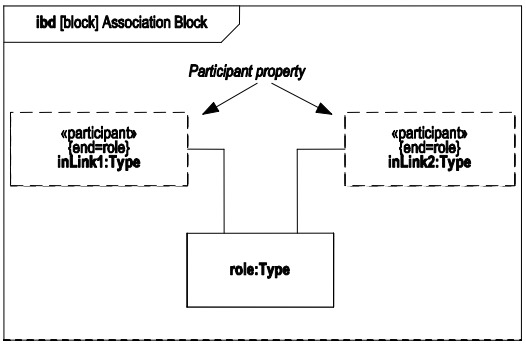
Appendix E

SysML overview (21)

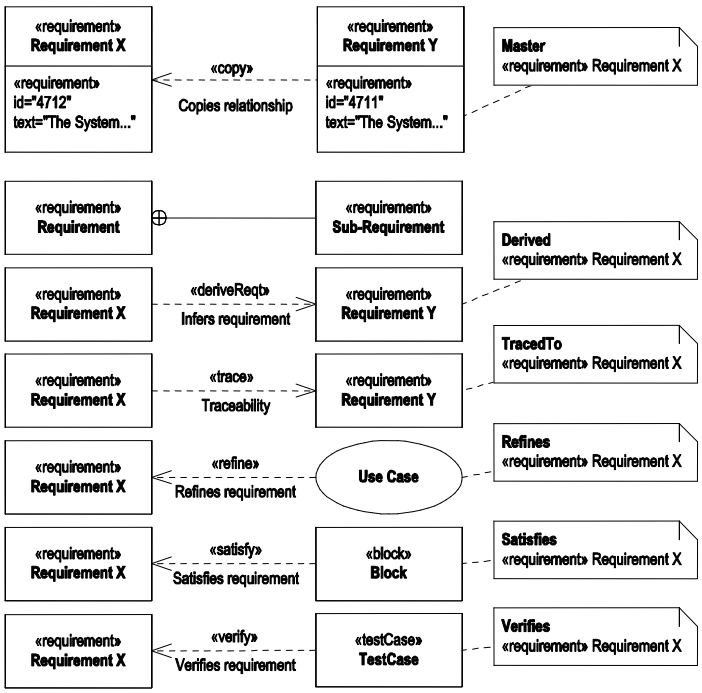


Internal Block Diagram





Requirements



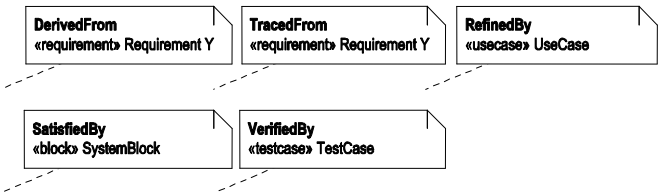
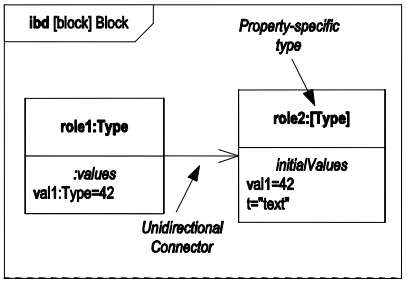
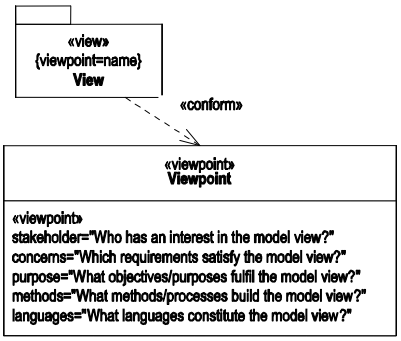


table Requirements Table

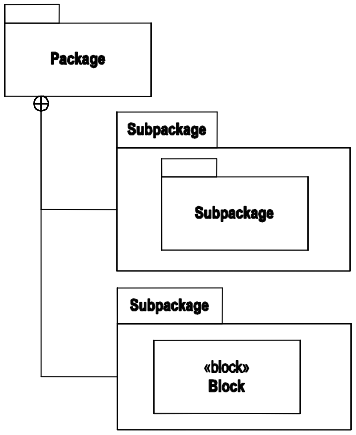
ID	Name	Text
4711	Requirement	The System...
...	...	...



Model View



Packages

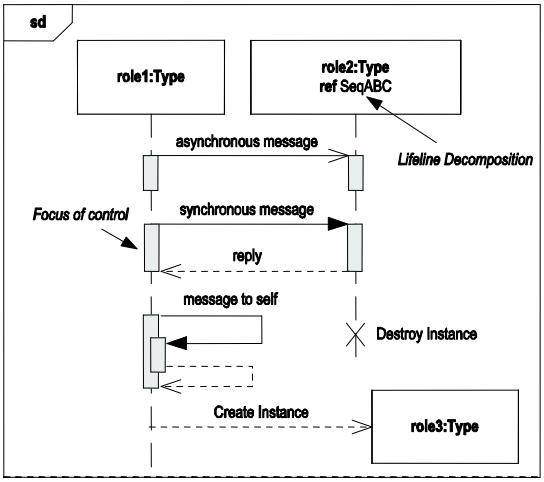


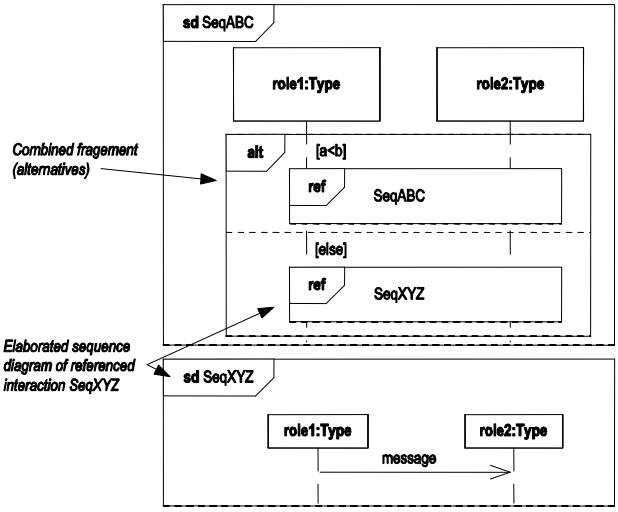
OMG Systems Modeling Language (OMG SysML™) 1.1 (http://www.omgsysml.org)
incl. SYSMOD stereotypes (http://www.sysmod.de)
Tim Weilkiens, Systems Engineering with SysML/UML (http://www.system-modeling.com)

Reference card, page 2/4

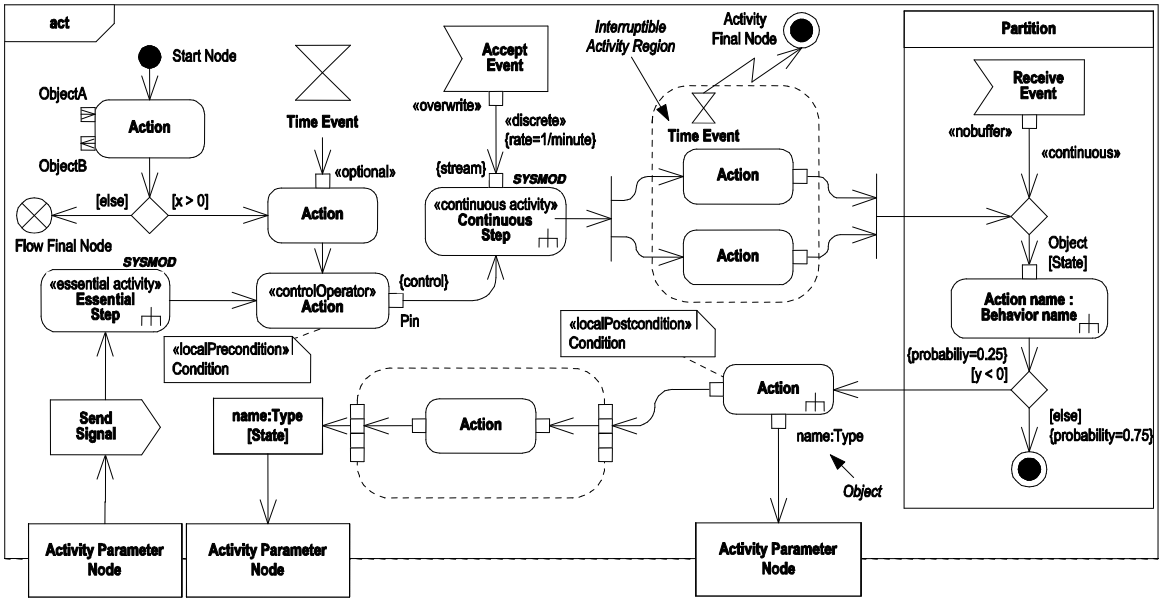
Translation support T. Netter

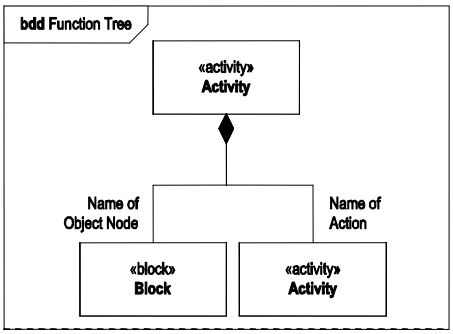
Sequence Diagram



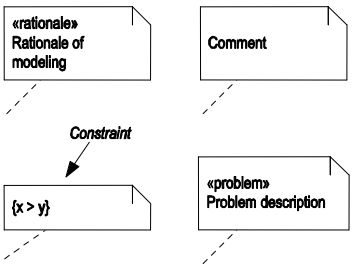


Activity Diagram





Comments

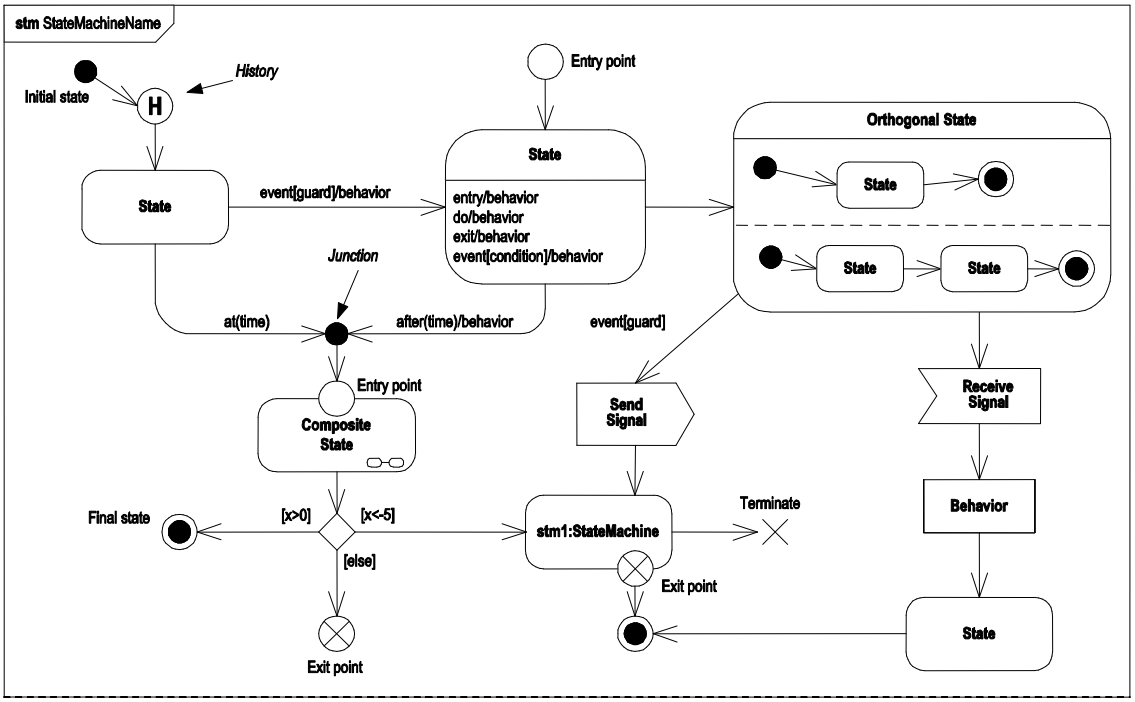


OMG Systems Modeling Language (OMG SysML™) 1.1 (http://www.omgsysml.org)
incl. SYSMOD stereotypes (http://www.sysmod.de)
Tim Weilkiens, *Systems Engineering with SysML/UML* (http://www.system-modeling.com)

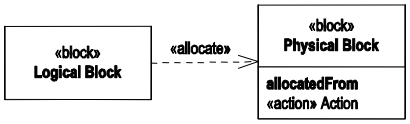
Reference card, page 3/4

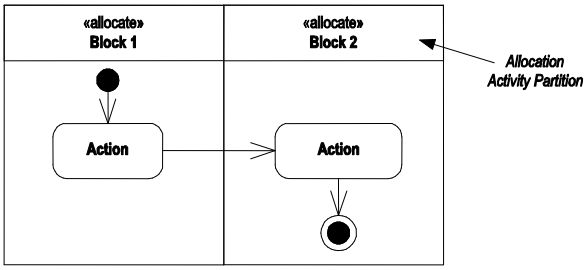
Translation support T. Netter

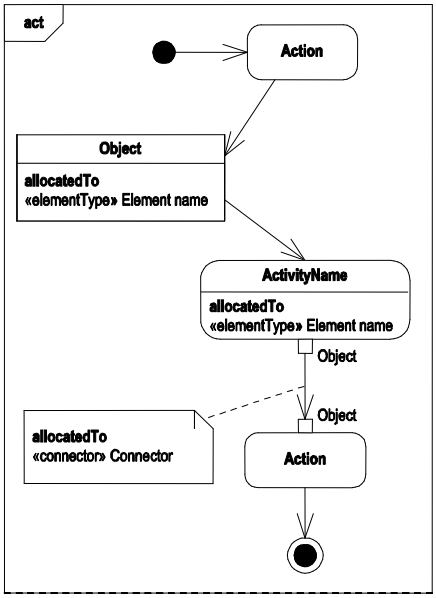
State Machines



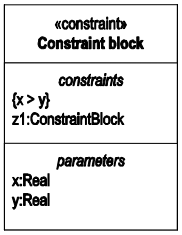
Allocations

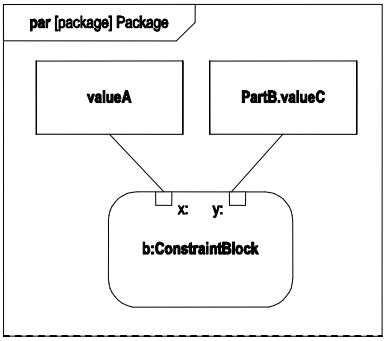






Parametric Diagram





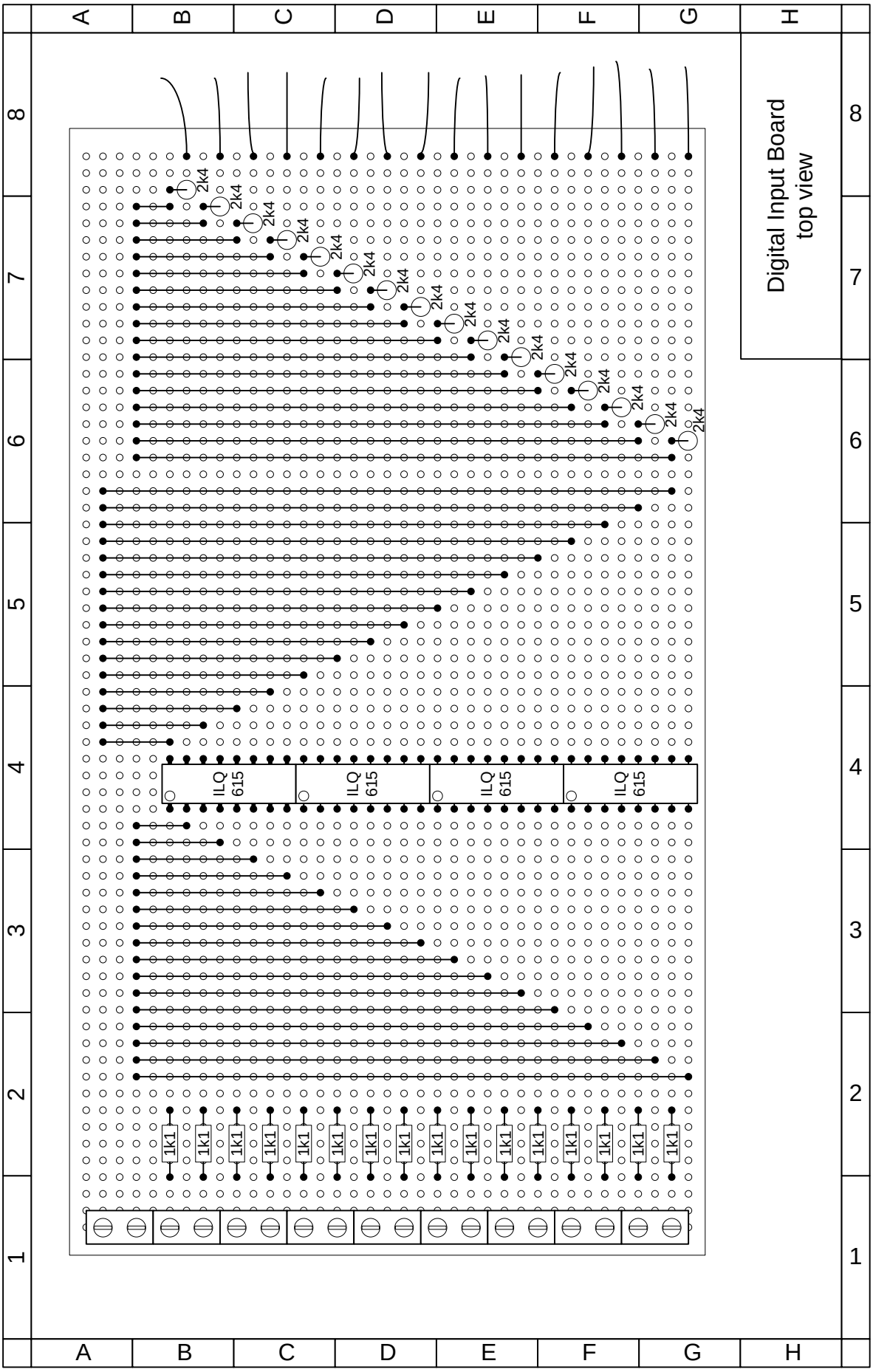
OMG Systems Modeling Language (OMG SysML™) 1.1 (http://www.omgsysml.org)
incl. SYSMOD stereotypes (http://www.sysmod.de)
Tim Weilkiens, *Systems Engineering with SysML/UML* (http://www.system-modeling.com)

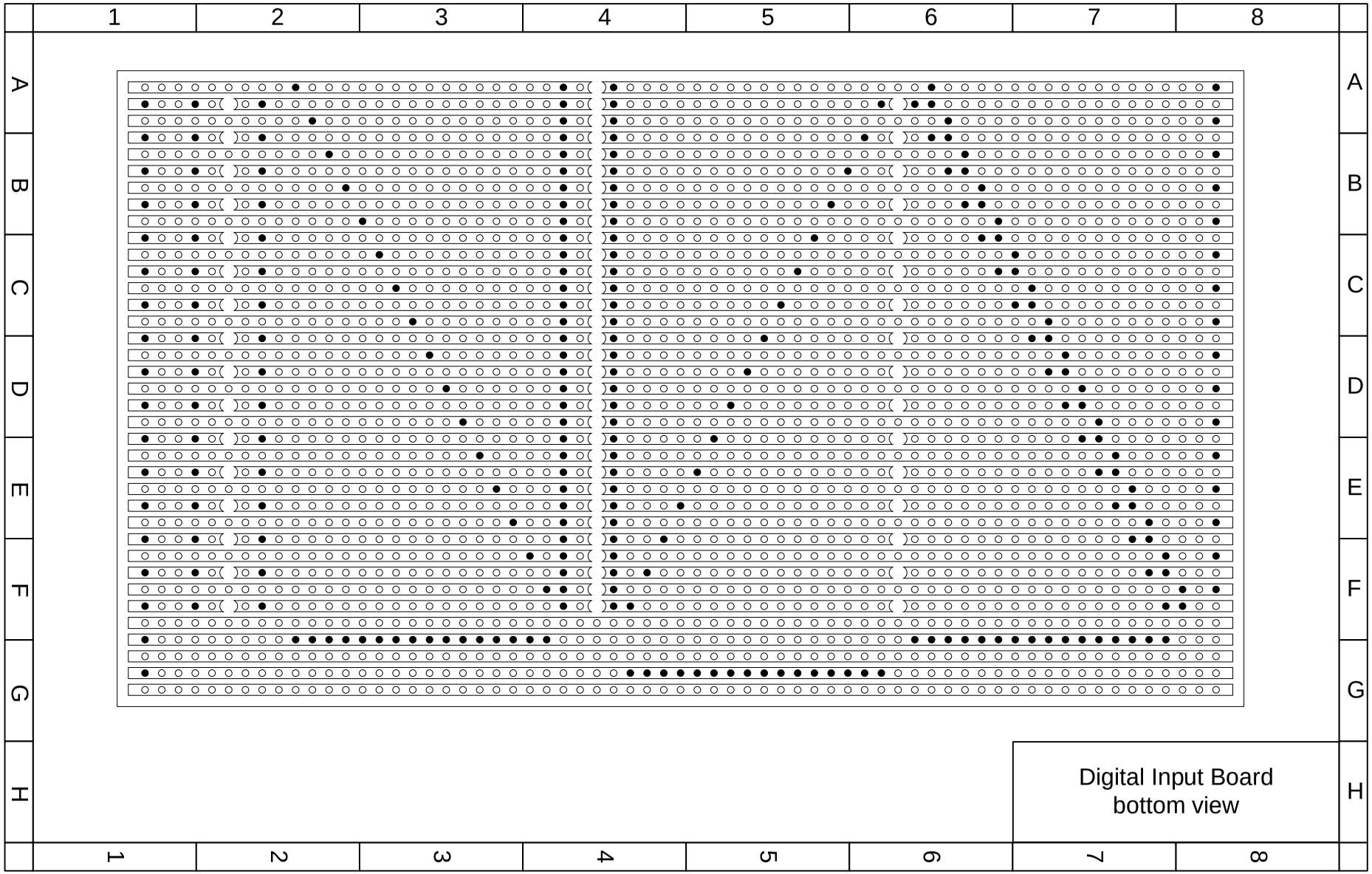
Reference card, page 4/4

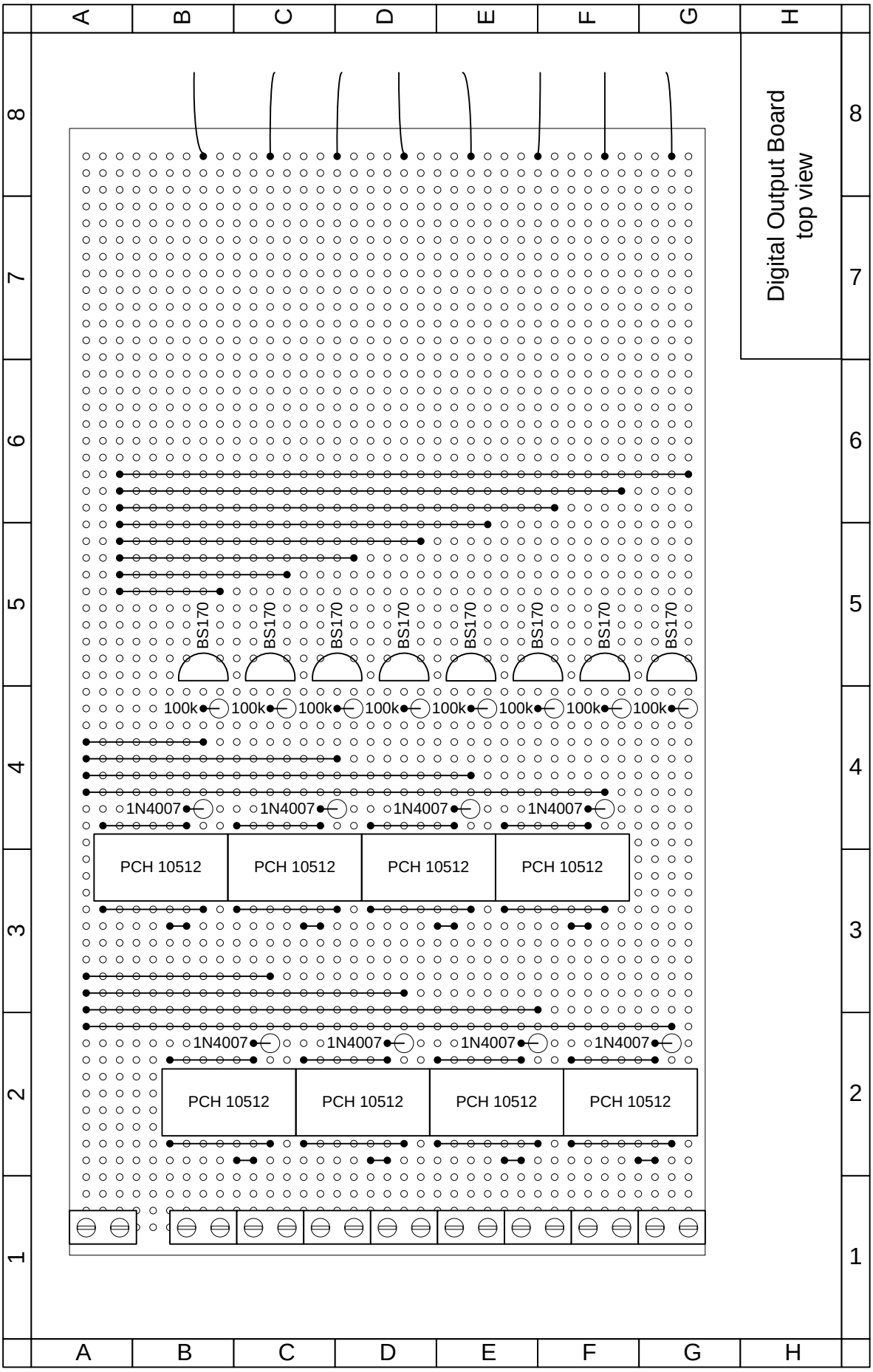
Translation support T. Netter

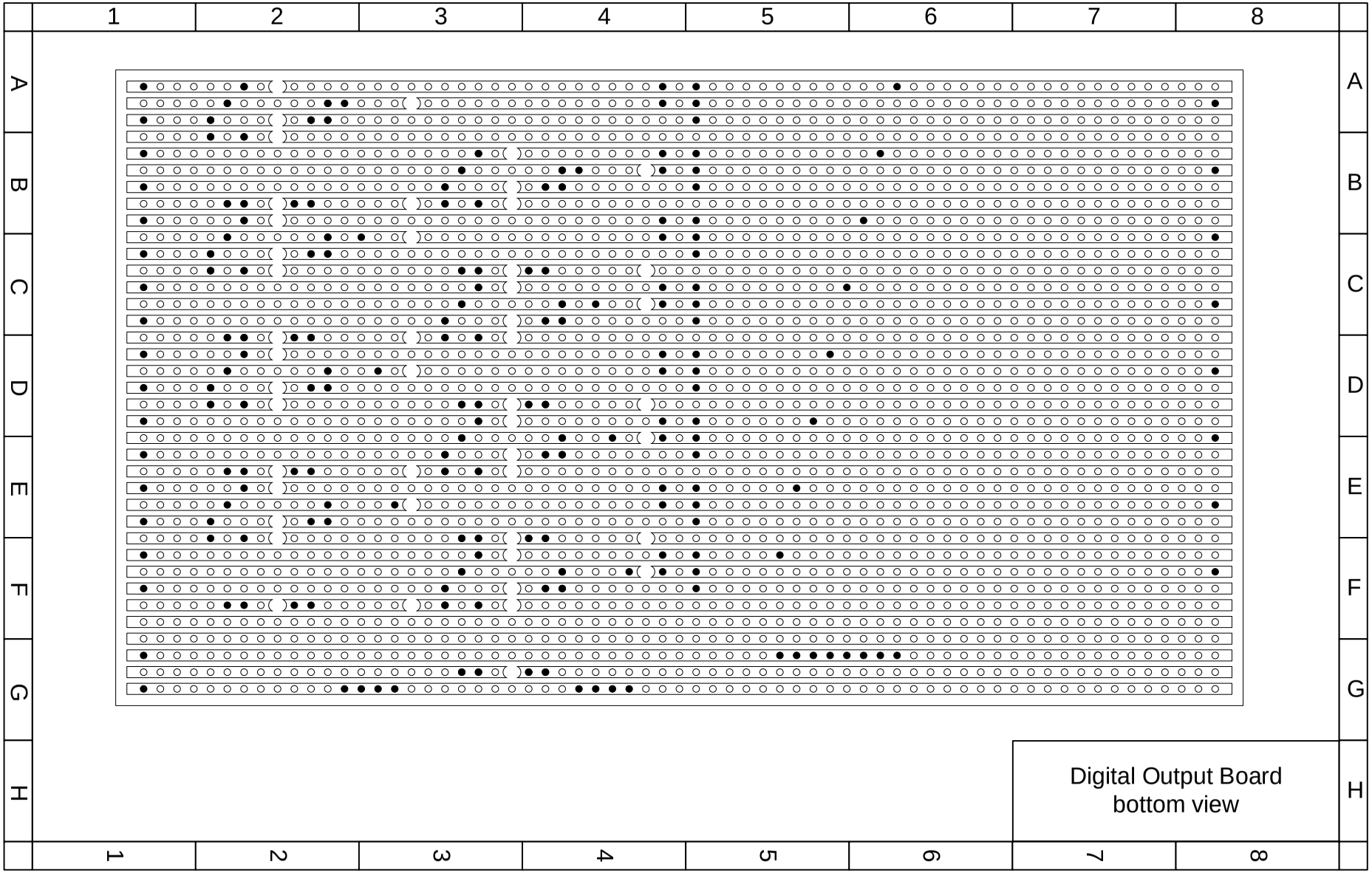
Appendix F

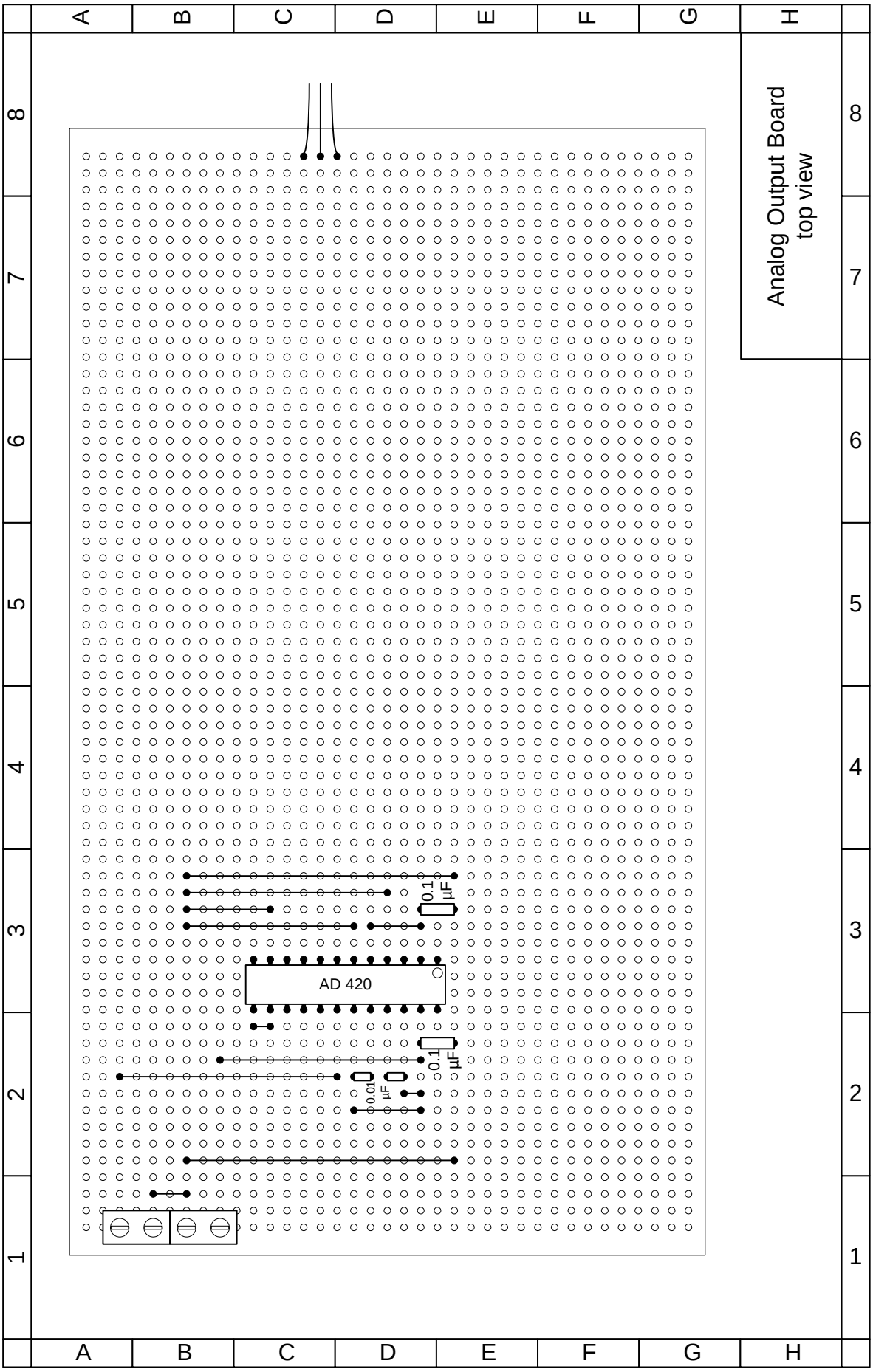
Interface Board Layouts

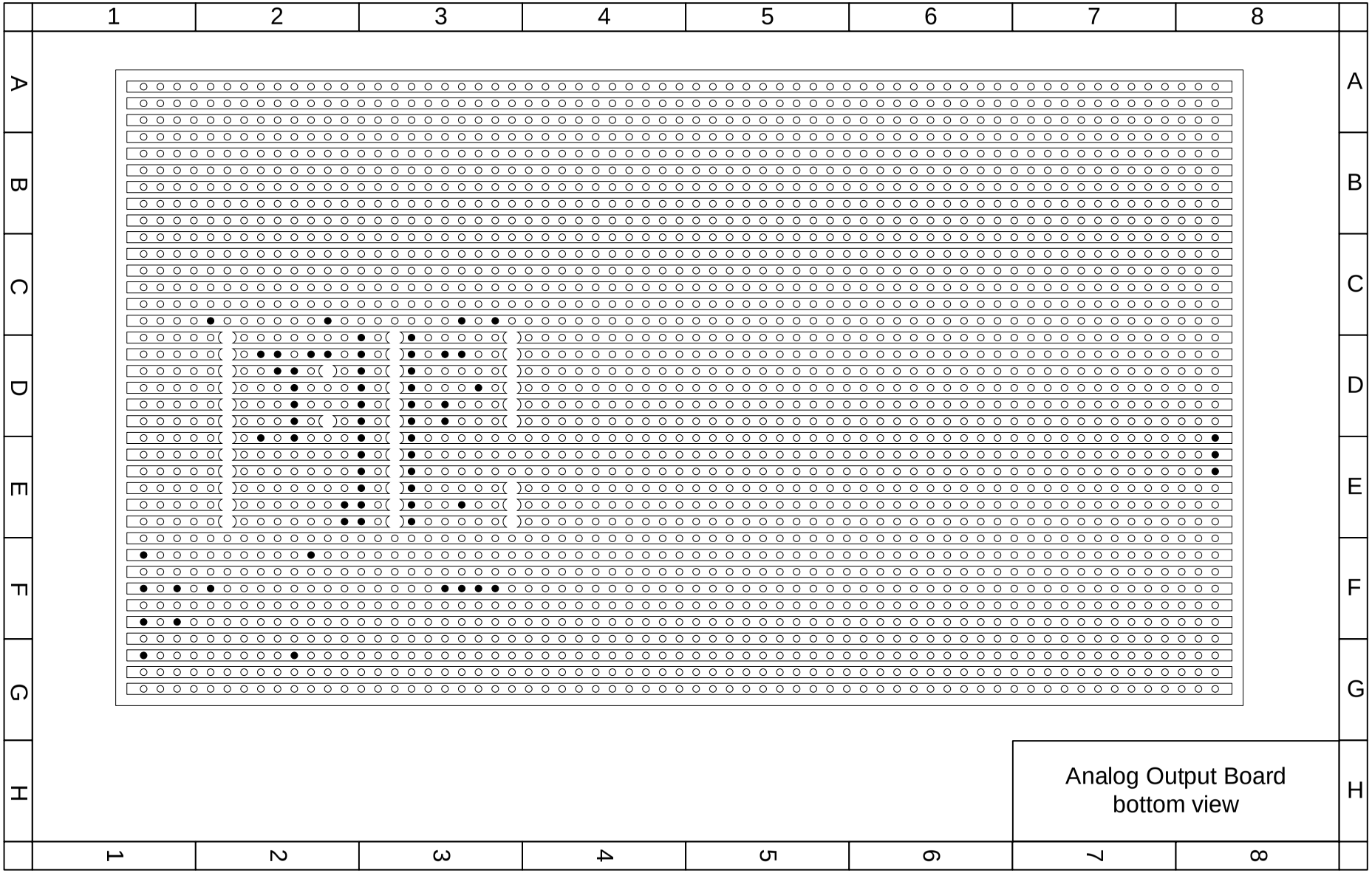

Digital Input Board
top view
ILQ 615
ILQ 615
ILQ 615
ILQ 615
1k1
2k4


1 2 3 4 5 6 7 8
A B C D E F G H
Digital Input Board
bottom view


Digital Output Board
top view
BS170
100k
1N4007
PCH 10512


1 2 3 4 5 6 7 8
A B C D E F G H
Digital Output Board
bottom view


Analog Output Board
top view
AD 420
0.1 µF
0.1 µF
0.01 µF


1 2 3 4 5 6 7 8
A B C D E F G H
Analog Output Board
bottom view

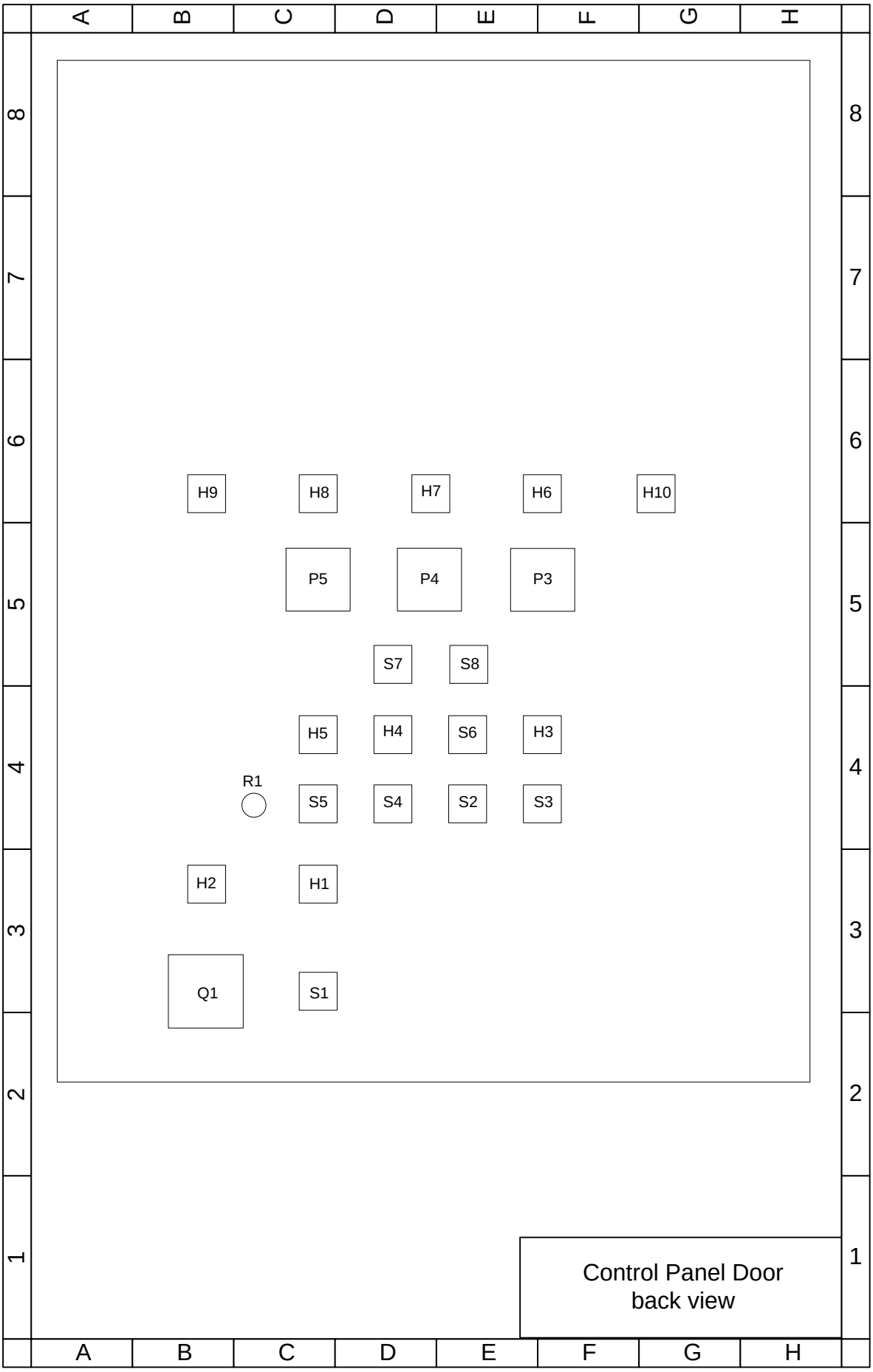
Appendix G

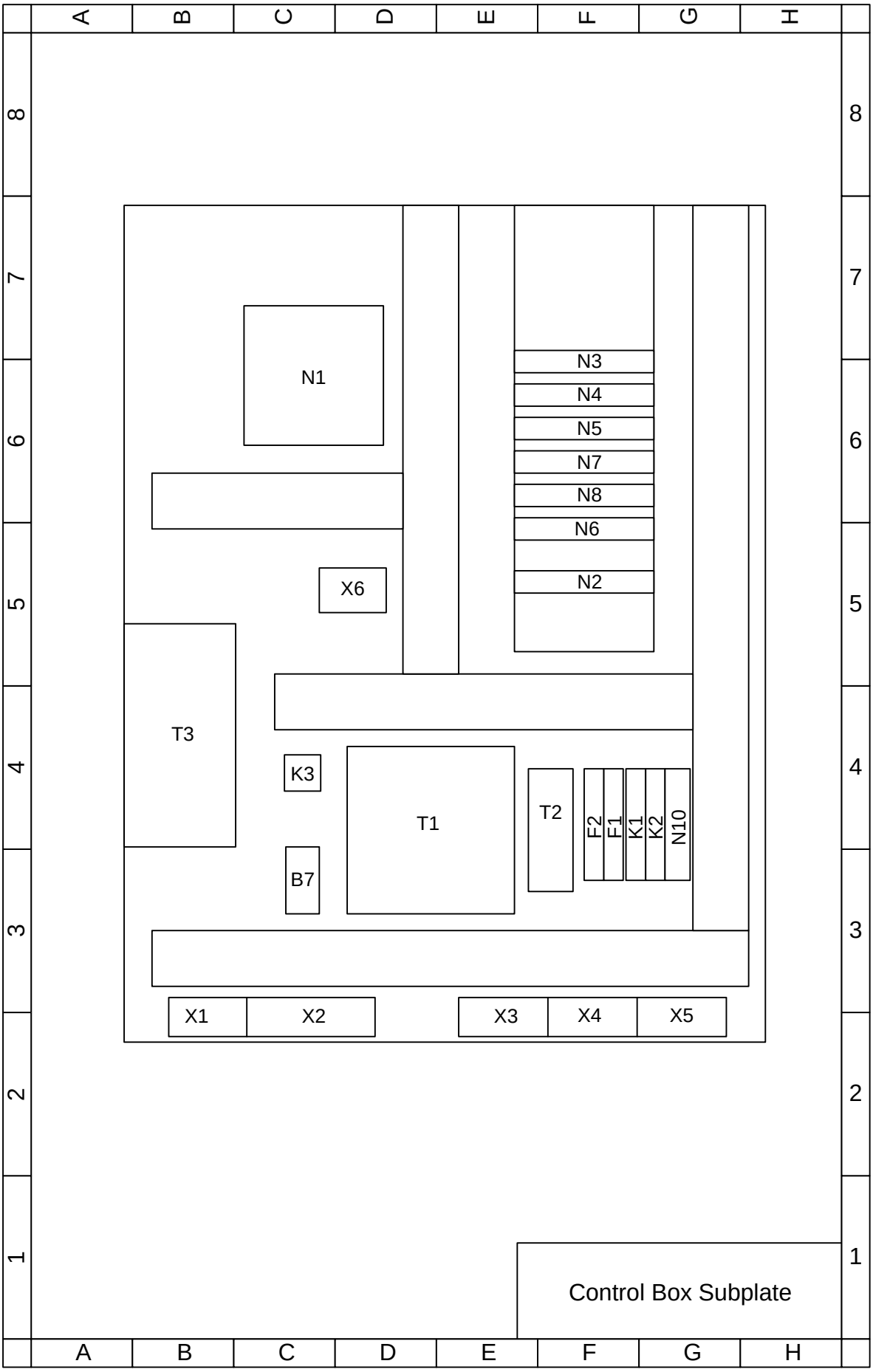
Schematics

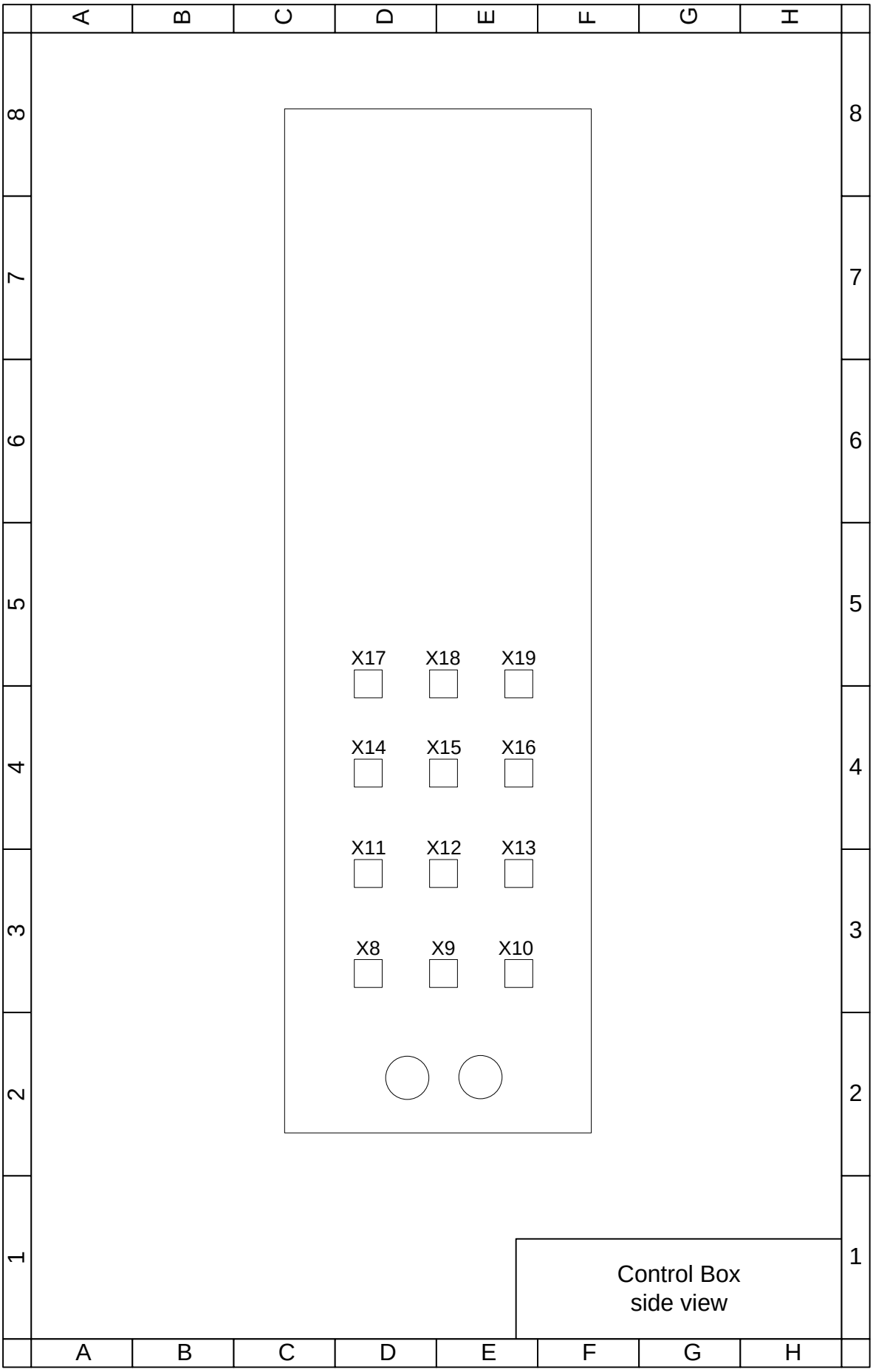
Code Letter	Description
B1	Air pressure sensor
B2	Air flow rate senor
B3	Air temperature sensor
B4	Fuel pressure sensor
B5	Fuel flow rate sensor
B6	Powder pressure sensor
B7	Glow plug current sensor
E1	Glow plug
E2	Powder feeder
E3	Air dryer
F1	Fuse for 24V power supply
F2	Fuse for 12V power supply
F3	Fuse for 110V transformer
H1	Emergency Stop light
H2	Controller ready light
H3	Auto light
H4	Manual light
H5	Powder feeding light
H6	Air error light
H7	Fuel error light
H8	Powder error light
H9	Glow plug error light
H10	No combustion light
K1	Fuel pump relay
K2	Powder feeder relay
K3	Glow plug relay
M1	Fuel pump
N1	Multi I/O Board
N2	Digital input board
N3	Digital output board 1
N4	Digital output board 2

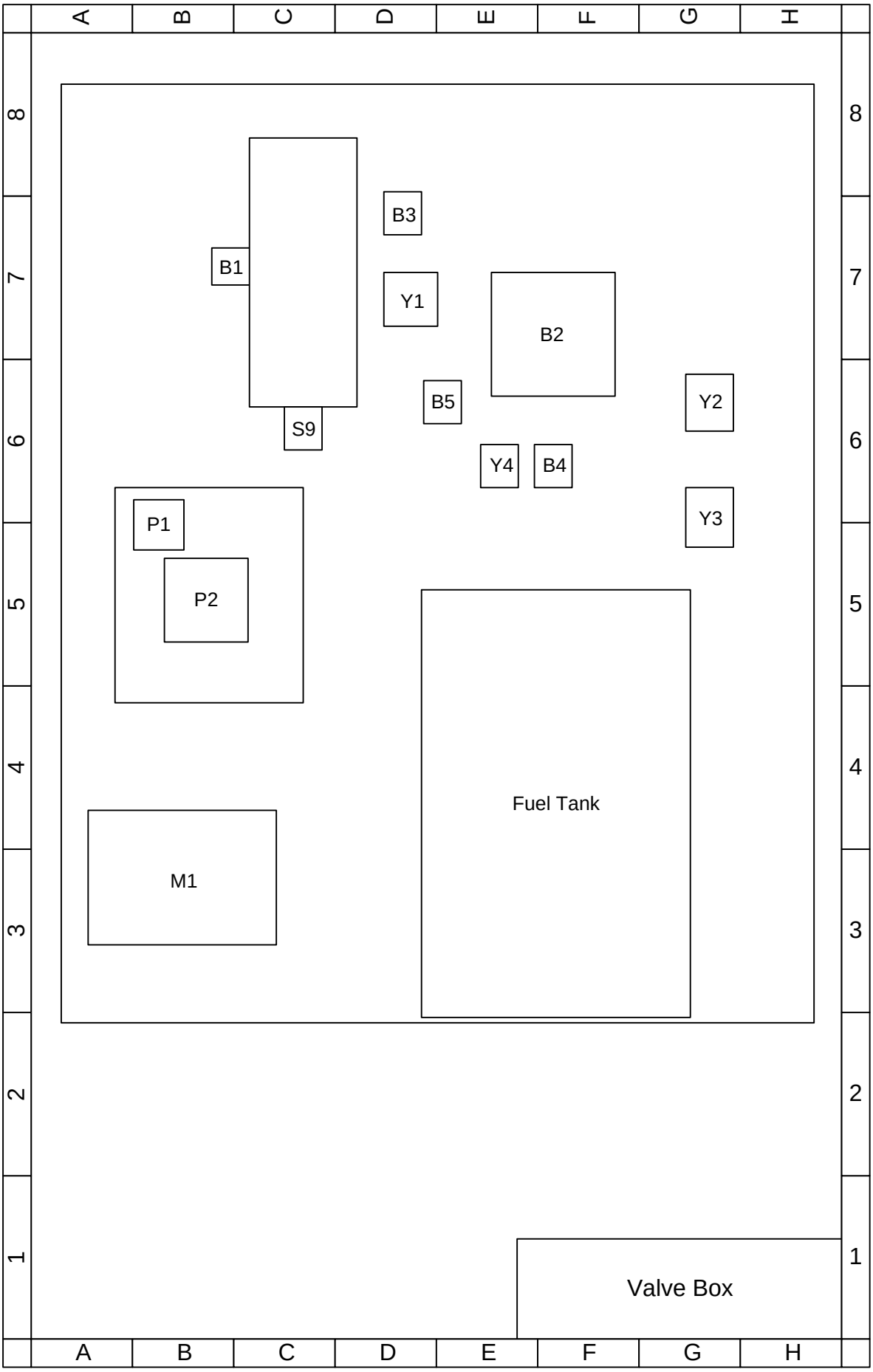
Code Letter	Description
N5	Digital output board 3
N6	Analog output board
N7	Analog input board 1
N8	Analog input board 2
N9	Amplifier for temperature sensor
N10	Controller for fuel proportional valve
N11	SRAM board
P1	Air flow rate display valve box
P2	Fuel flow rate display valve box
P3	Air display
P4	Fuel display
P5	Powder display
Q1	Main switch
R1	Potentiometer
S1	Emergency Stop button
S2	Reset button
S3	Auto button
S4	Manual button
S5	Feed button
S6	Clean switch
S7	Combustion start button
S8	Combustion stop button
S9	Air switch
T1	PC power supply
T2	24 V power supply
T3	12 V power supply
T4	110 V Transformer
V1	Free-wheeling diode on K1
V2	Free-wheeling diode on K2
V3	Free-wheeling diode on Y1
V4	Free-wheeling diode on Y2
V5	Free-wheeling diode on Y3
X1	Earth terminals
X2	230 V AC terminals

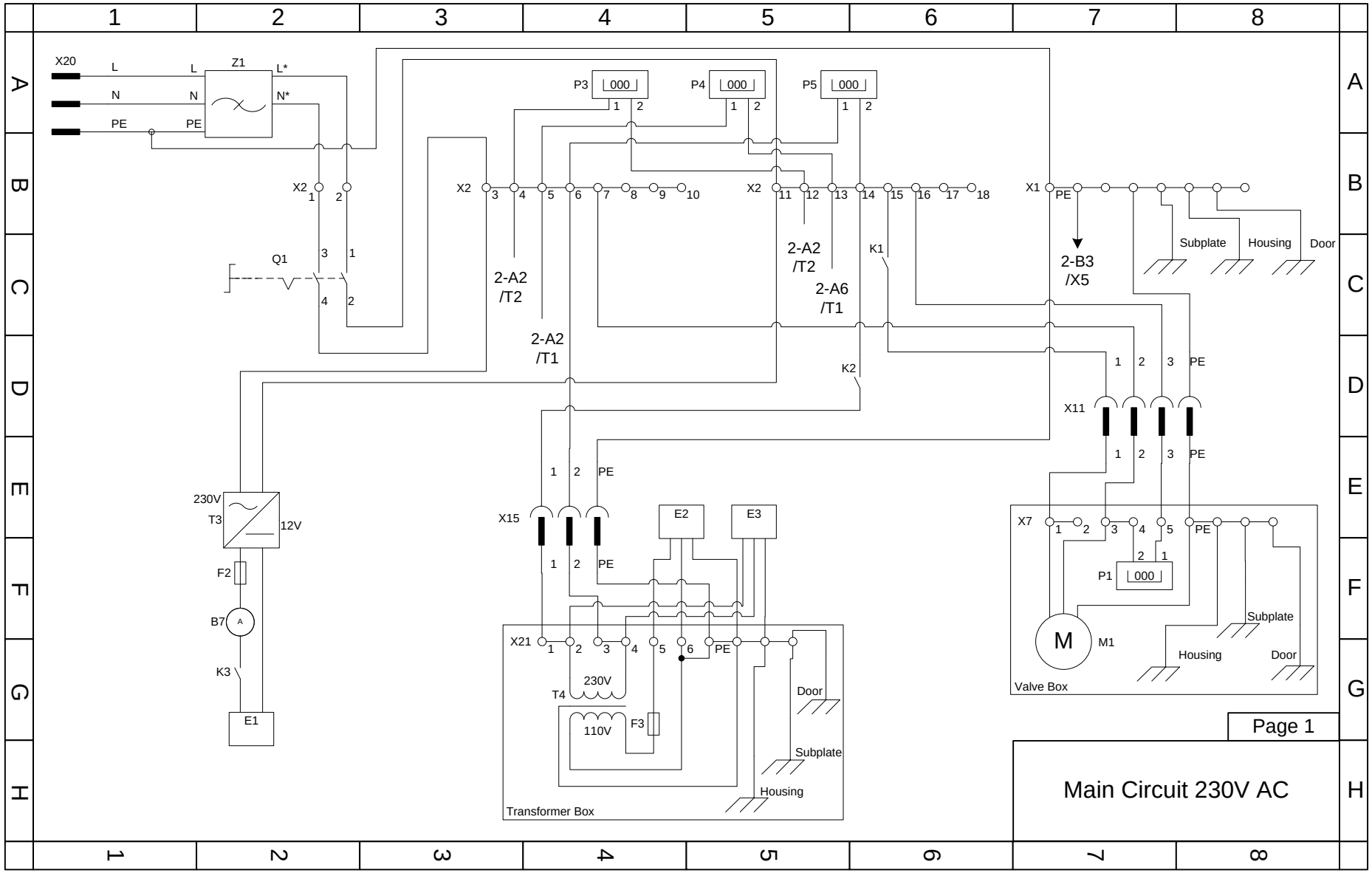
Code Letter	Description
X3	12 V DC terminals
X4	24 V DC terminals
X5	Ground terminals
X6	PC power supply terminals
X7	Valve box terminals
X8	Fuel control plug
X9	Fuel valves plug
X10	Fuel sensors plug
X11	Fuel pump plug
X12	Air switch/valve plug
X13	Air sensors plug
X14	Spare plug
X15	Powder feeder supply plug
X16	Air temperature plug
X17	Spare Plug
X18	Powder feeder start plug
X19	Powder sensors plug
X20	Mains plug
X21	Transformer box terminals
X22	RS232 plug
Y1	Main air 2/3 valve
Y2	Main fuel 2/3 valve
Y3	Ignition fuel 2/3 valve
Y4	Fuel proportional valve
Z1	230 V AC line filter

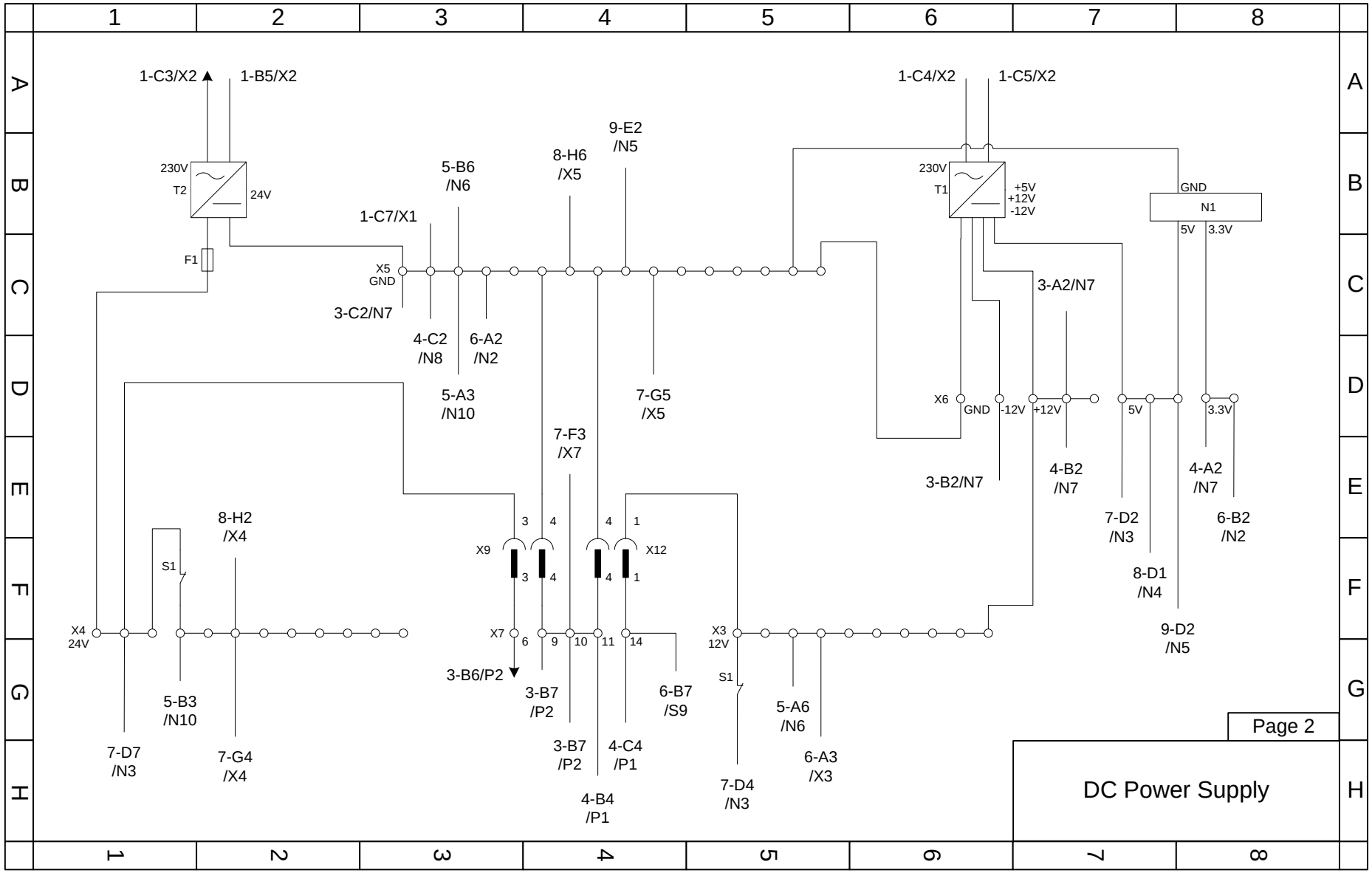

A B C D E F G H
8 7 6 5 4 3 2 1
H9
H8
H7
H6
H10
P5
P4
P3
S7
S8
H5
H4
S6
H3
R1
S5
S4
S2
S3
H2
H1
Q1
S1
Control Panel Door
back view

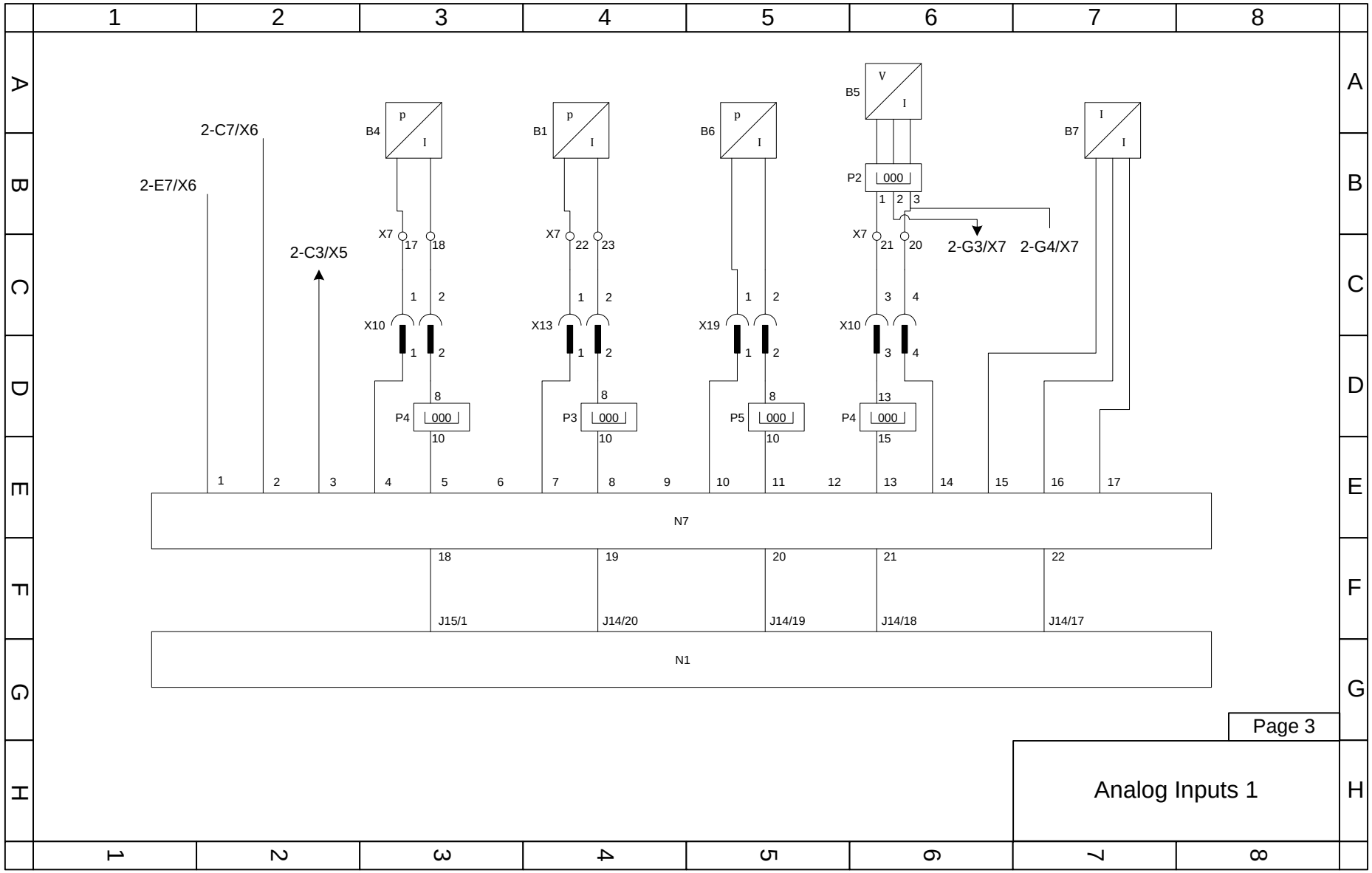

A
B
C
D
E
F
G
H
8
7
6
5
4
3
2
1
N1
N3
N4
N5
N7
N8
N6
N2
X6
T3
K3
T1
T2
F2
F1
K1
K2
N10
B7
X1
X2
X3
X4
X5
Control Box Subplate

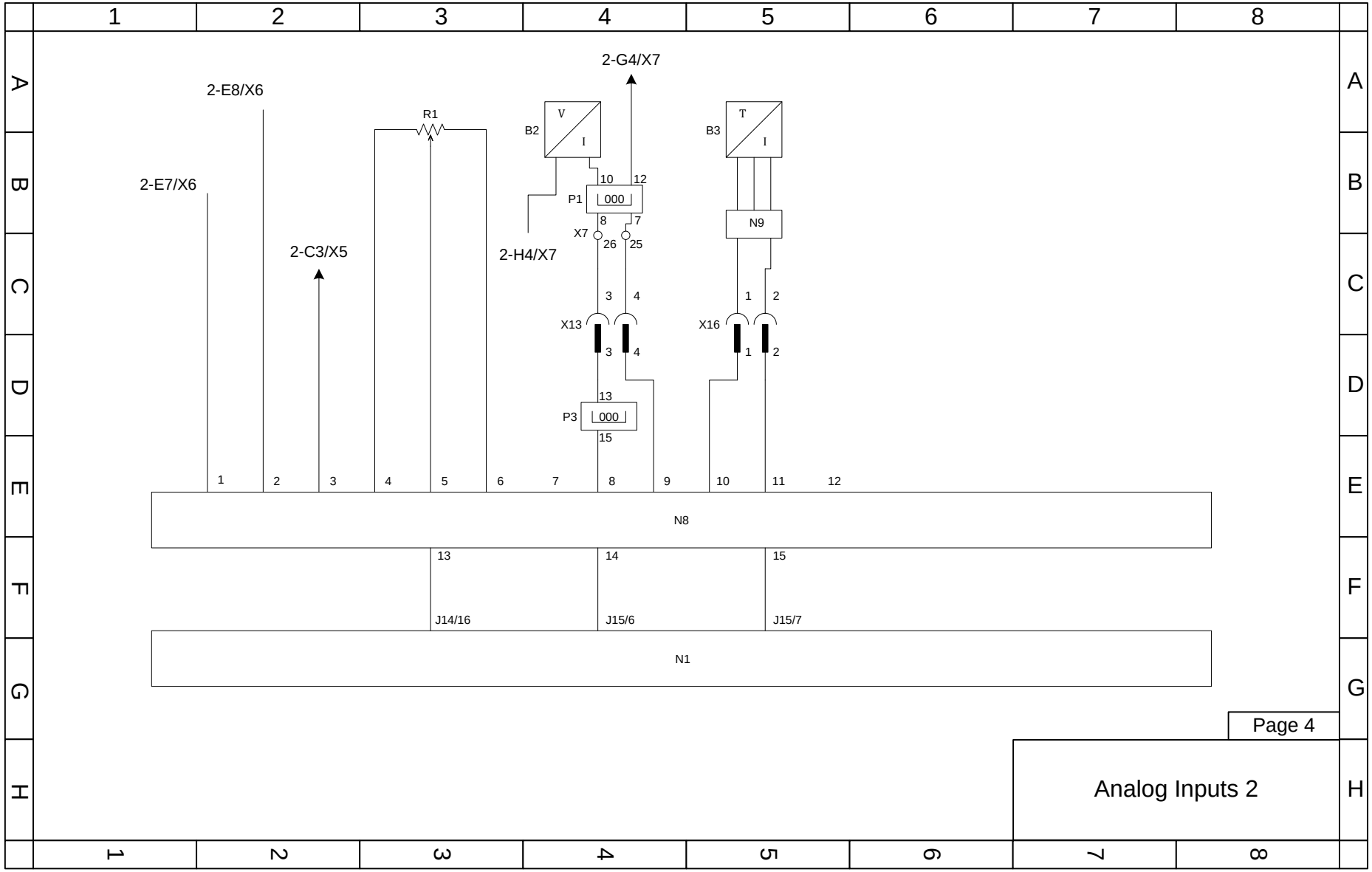

A B C D E F G H
8 7 6 5 4 3 2 1
X17 X18 X19
X14 X15 X16
X11 X12 X13
X8 X9 X10
Control Box
side view

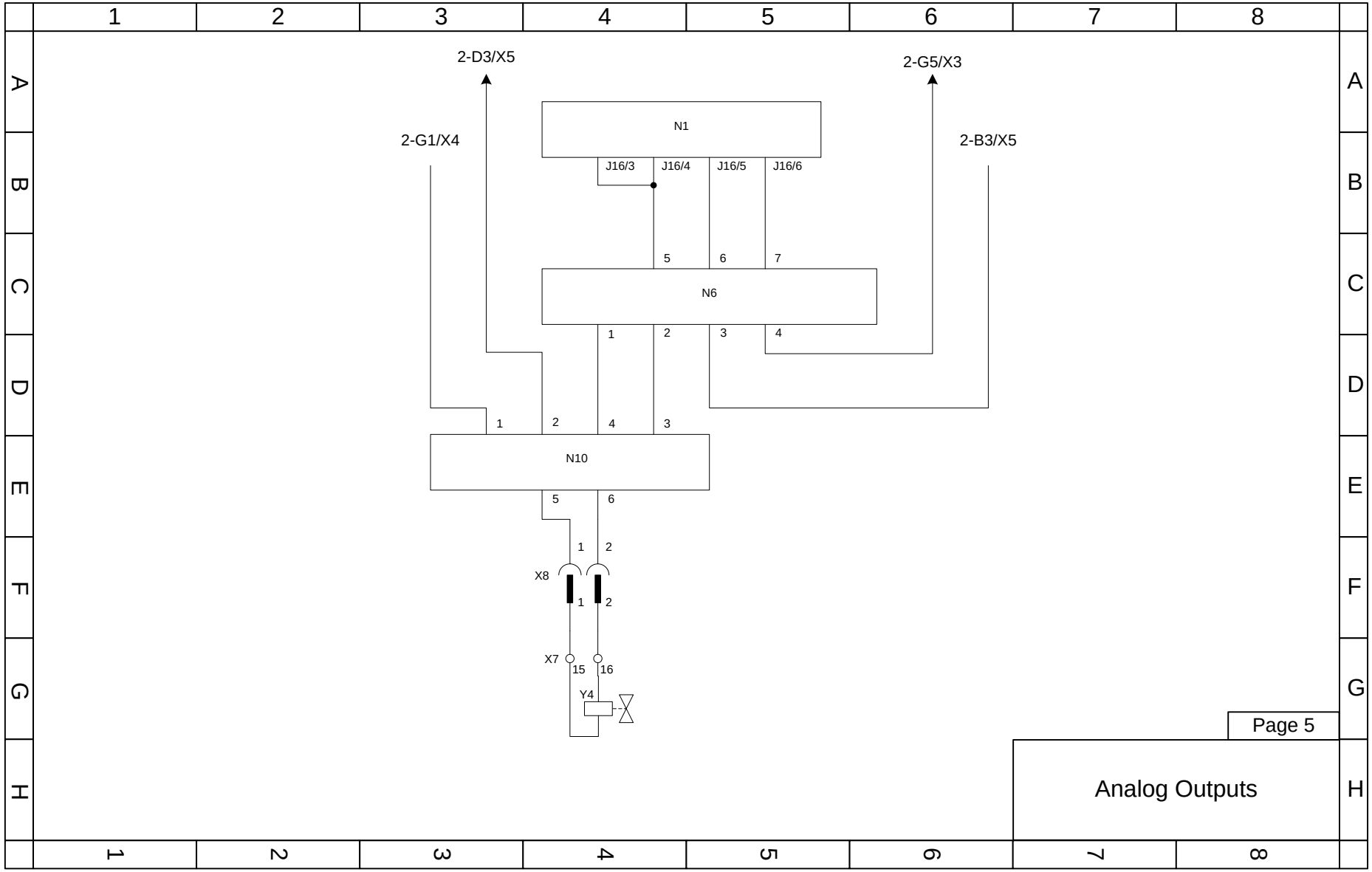

A
B
C
D
E
F
G
H
8
7
6
5
4
3
2
1
B1
S9
B3
Y1
B2
B5
Y2
Y4
B4
Y3
P1
P2
Fuel Tank
M1
Valve Box

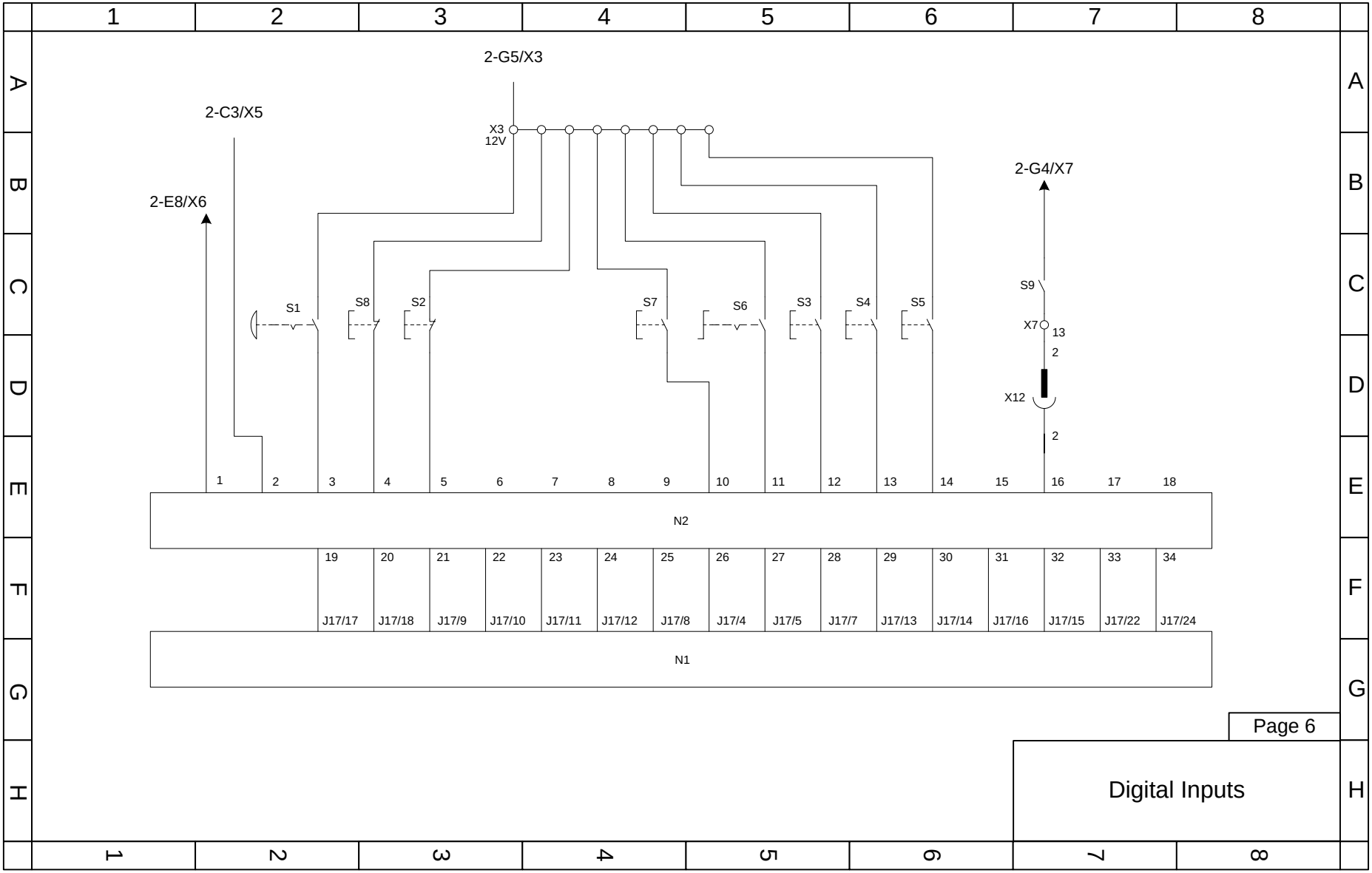

X20
L
N
PE
Z1
L*
N*
P3
P4
P5
000
X2
X1
Q1
2-A2
/T2
2-A2
/T1
2-A2
/T2
2-A6
/T1
K1
K2
2-B3
/X5
Subplate
Housing
Door
X11
X15
230V
T3
12V
F2
B7
K3
E1
E2
E3
X21
T4
110V
F3
Transformer Box
X7
P1
M
M1
Valve Box
Page 1
Main Circuit 230V AC

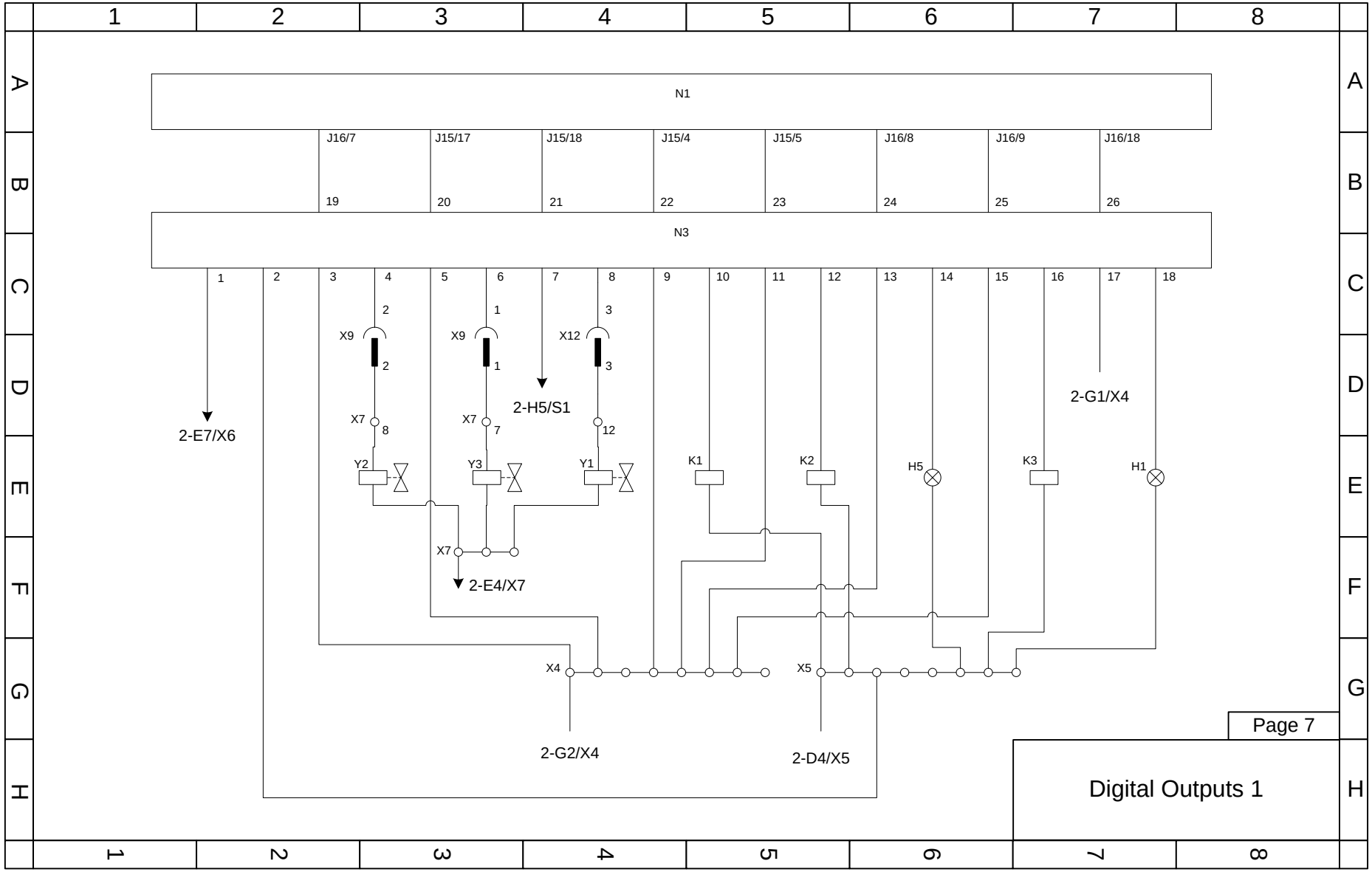

1-C3/X2
1-B5/X2
1-C4/X2
1-C5/X2
230V
T2
24V
F1
230V
T1
+5V
+12V
-12V
GND
N1
5V
3.3V
1-C7/X1
5-B6
/N6
8-H6
/X5
9-E2
/N5
X5
GND
3-C2/N7
4-C2
/N8
6-A2
/N2
5-A3
/N10
7-G5
/X5
3-A2/N7
X6
GND
-12V
+12V
5V
3.3V
3-B2/N7
4-B2
/N7
4-A2
/N7
7-D2
/N3
6-B2
/N2
8-D1
/N4
9-D2
/N5
7-F3
/X7
8-H2
/X4
S1
X9
X12
X4
24V
X7
X3
12V
3-B6/P2
3-B7
/P2
6-B7
/S9
5-B3
/N10
7-D7
/N3
7-G4
/X4
3-B7
/P2
4-C4
/P1
4-B4
/P1
S1
5-A6
/N6
6-A3
/X3
7-D4
/N3
Page 2
DC Power Supply

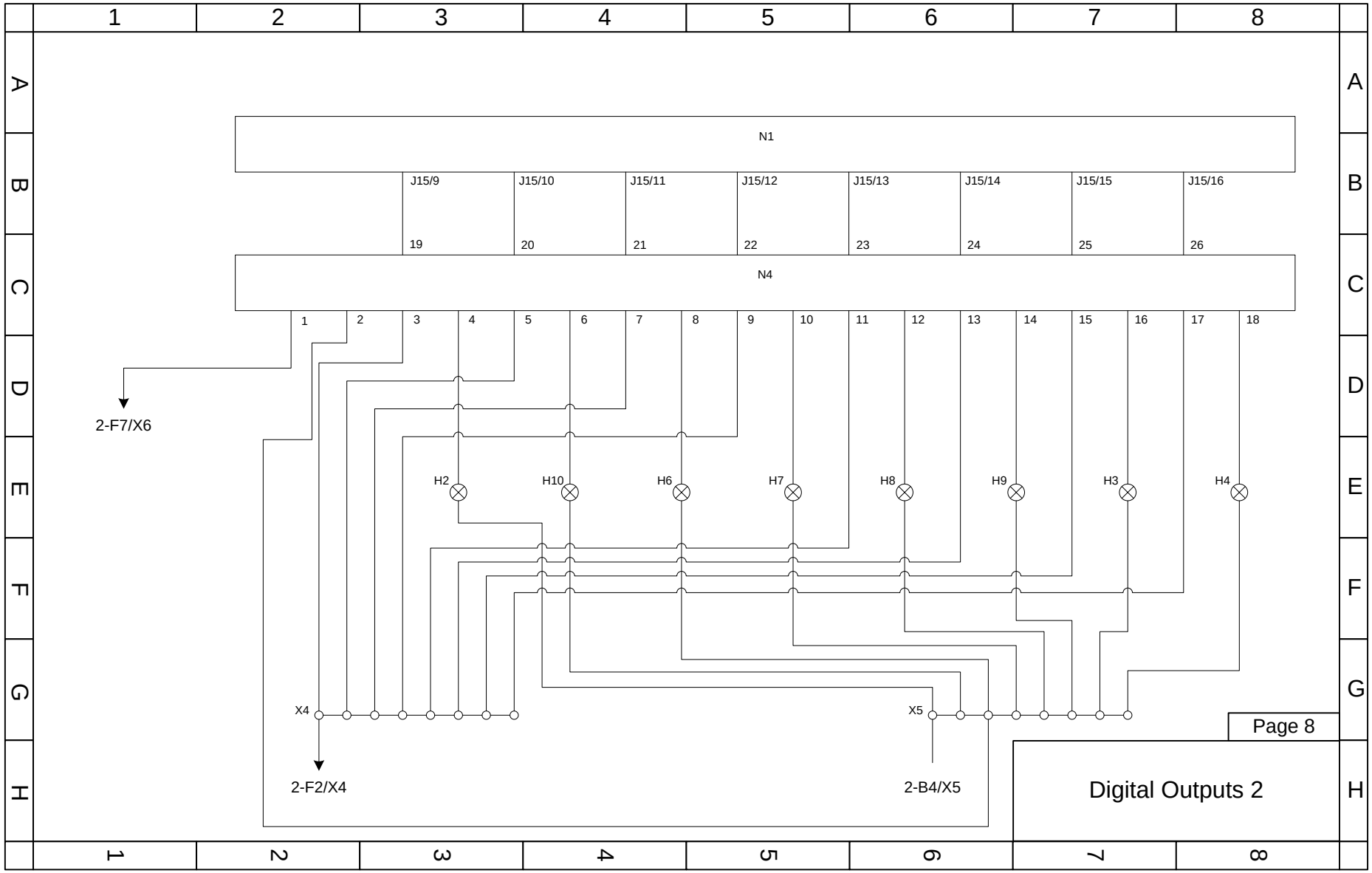

2-E7/X6
2-C7/X6
2-C3/X5
B4
B1
B6
B5
B7
P2
X7
X10
X13
X19
P4
P3
P5
2-G3/X7
2-G4/X7
N7
N1
J15/1
J14/20
J14/19
J14/18
J14/17
Page 3
Analog Inputs 1

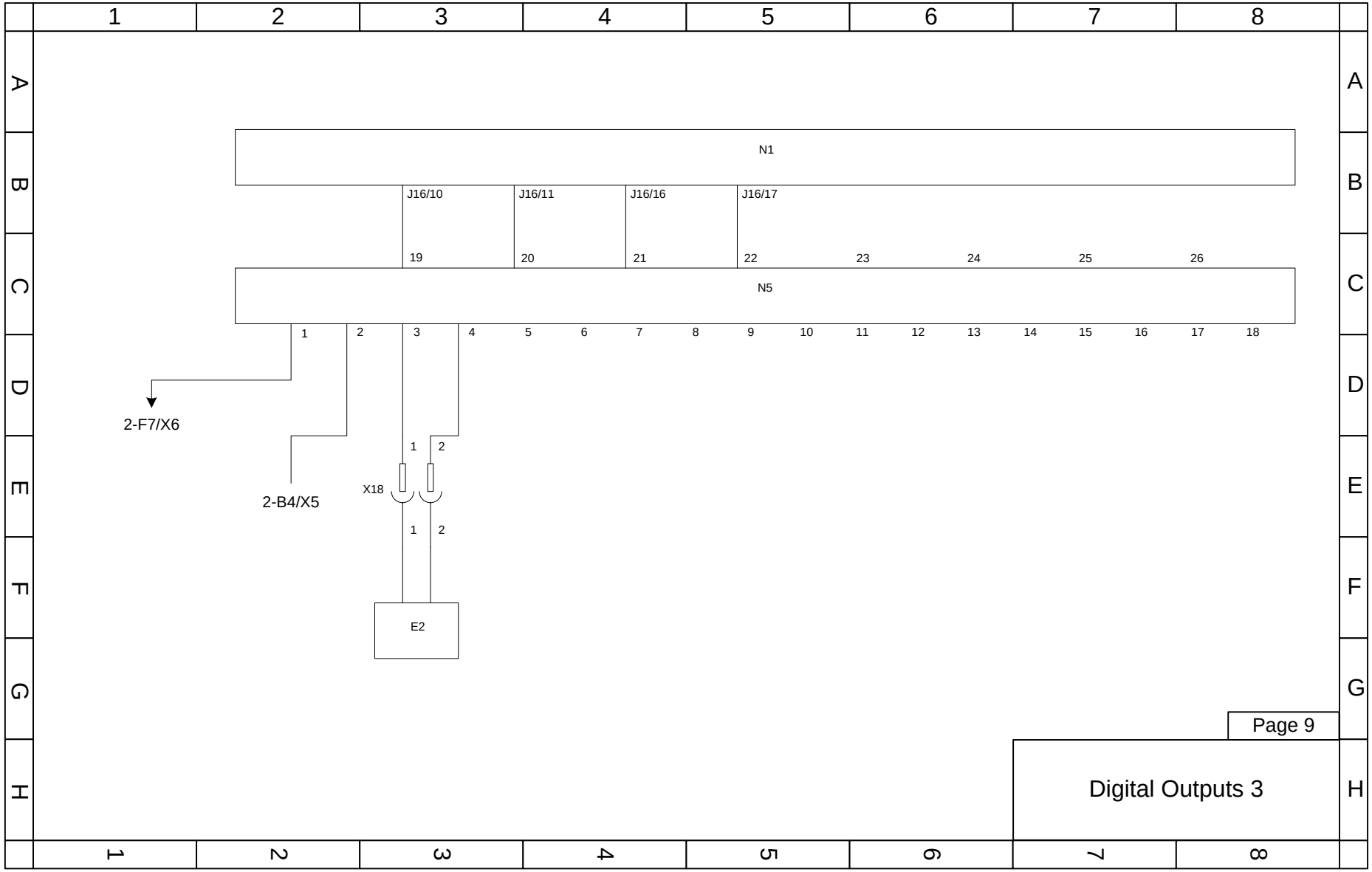

2-E8/X6
2-E7/X6
2-C3/X5
R1
B2
V
I
2-G4/X7
2-H4/X7
P1
000
X7
B3
T
I
N9
X13
X16
P3
000
N8
J14/16
J15/6
J15/7
N1
Page 4
Analog Inputs 2

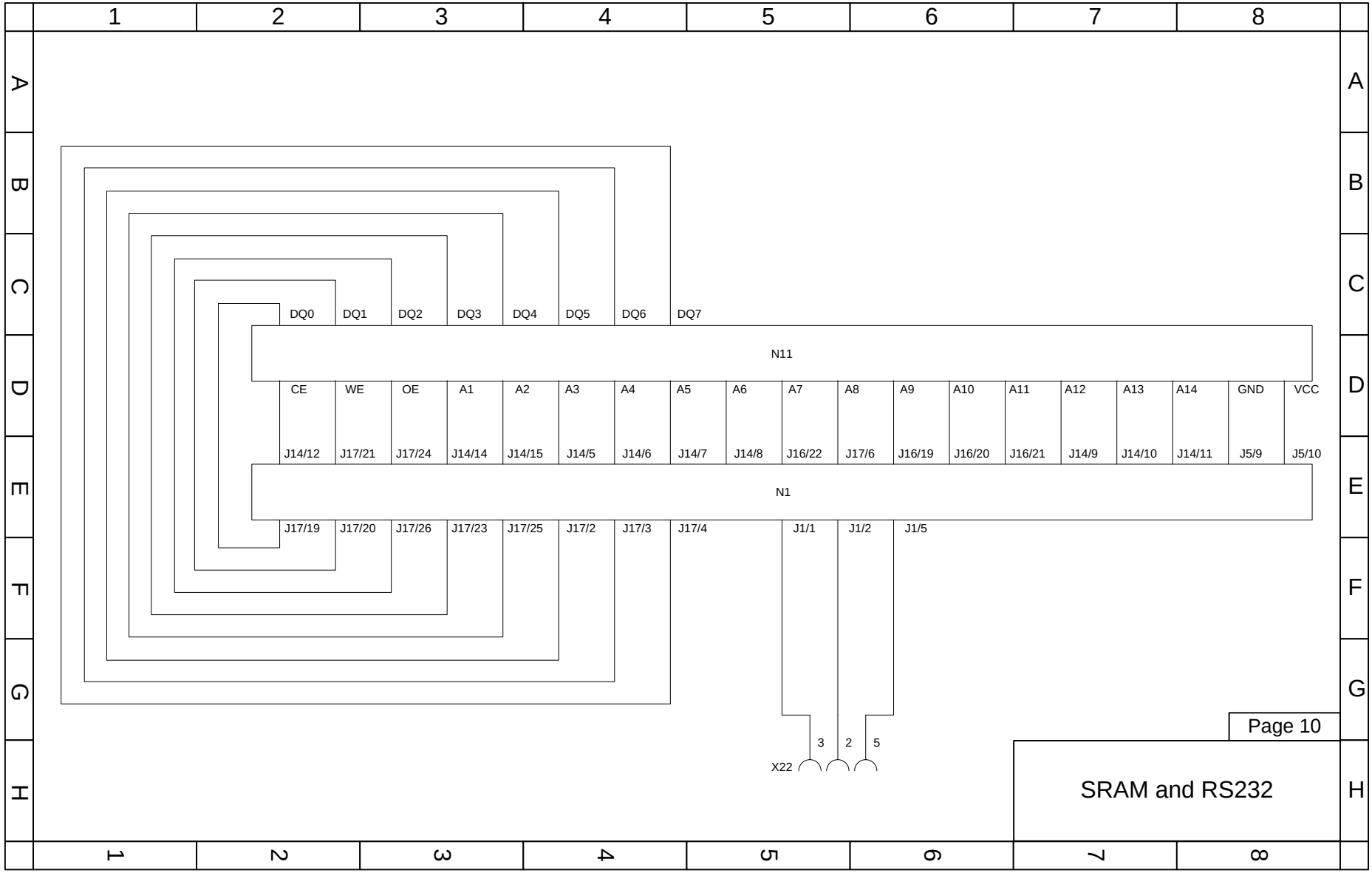

2-D3/X5
2-G5/X3
2-G1/X4
2-B3/X5
N1
J16/3
J16/4
J16/5
J16/6
5
6
7
N6
1
2
3
4
1
2
4
3
N10
5
6
1
2
X8
1
2
X7
15
16
Y4
Page 5
Analog Outputs


2-G5/X3
X3
12V
2-C3/X5
2-E8/X6
2-G4/X7
S1
S8
S2
S7
S6
S3
S4
S5
S9
X7
13
2
X12
2
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
N2
19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
J17/17 J17/18 J17/9 J17/10 J17/11 J17/12 J17/8 J17/4 J17/5 J17/7 J17/13 J17/14 J17/16 J17/15 J17/22 J17/24
N1
Page 6
Digital Inputs


N1
J16/7
J15/17
J15/18
J15/4
J15/5
J16/8
J16/9
J16/18
19
20
21
22
23
24
25
26
N3
X9
X9
X12
X7
X7
Y2
Y3
Y1
K1
K2
H5
K3
H1
2-E7/X6
2-H5/S1
2-G1/X4
2-E4/X7
X4
X5
2-G2/X4
2-D4/X5
Page 7
Digital Outputs 1


N1
J15/9
J15/10
J15/11
J15/12
J15/13
J15/14
J15/15
J15/16
19
20
21
22
23
24
25
26
N4
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
2-F7/X6
H2
H10
H6
H7
H8
H9
H3
H4
X4
X5
2-F2/X4
2-B4/X5
Page 8
Digital Outputs 2


N1
J16/10
J16/11
J16/16
J16/17
19
20
21
22
23
24
25
26
N5
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
2-F7/X6
2-B4/X5
X18
1 2
1 2
E2
Page 9
Digital Outputs 3


DQ0
DQ1
DQ2
DQ3
DQ4
DQ5
DQ6
DQ7
N11
CE
WE
OE
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
GND
VCC
J14/12
J17/21
J17/24
J14/14
J14/15
J14/5
J14/6
J14/7
J14/8
J16/22
J17/6
J16/19
J16/20
J16/21
J14/9
J14/10
J14/11
J5/9
J5/10
N1
J17/19
J17/20
J17/26
J17/23
J17/25
J17/2
J17/3
J17/4
J1/1
J1/2
J1/5
3
2
5
X22
Page 10
SRAM and RS232