

Control over CAN and Flexray  
**Embedded Control Systems**

Sai Krishna Kalluri & Snorri Stefansson

April 2, 2017

# Contents

<b>Introduction</b>	<b>2</b>
<b>1 Part 1</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Response Time analysis . . . . .	3
1.2.1 Response time analysis per processing unit . . . . .	3
1.2.2 Response time analysis for the CAN bus messages . . . . .	3
1.3 System model . . . . .	3
1.4 Design decision . . . . .	3
1.5 Results . . . . .	3
<b>2 Part 2</b>	<b>4</b>
2.1 Introduction . . . . .	4
2.2 Response Time analysis . . . . .	5
2.2.1 Response time analysis per processing unit . . . . .	5
2.2.2 Response time analysis for the CAN bus messages . . . . .	6
2.3 Optimisation for sensor-to-actuator delay . . . . .	6
2.4 System model . . . . .	6
2.5 Design decision . . . . .	6
2.6 Results . . . . .	6
2.7 Conclusions . . . . .	7
<b>3 Part 3</b>	<b>8</b>
3.1 Introduction . . . . .	8
3.2 Answer all the questions . . . . .	8
3.2.1 Theoretical analysis versus actual implementation . . . . .	8
3.3 Design decision . . . . .	8
3.4 Results . . . . .	8
3.5 Conclusions . . . . .	8
3.6 Results . . . . .	8
3.7 Conclusion . . . . .	8

# Introduction

# Chapter 1

## Part 1

### 1.1 Introduction

### 1.2 Response Time analysis

#### 1.2.1 Response time analysis per processing unit

Table 1.1: By running the Matlab script `ResponsetimeAnylsis.FPP.m` with the different parameters given for PU1 and PU2 these response times are obtained for each of the tasks. These files are then delivered as PU1.m PU2.m

PU1	$T_1$	$T_2$	$T_3$	$T_4 (T_s)$
Matlab (ms)	0.1	2.1	4.1	7.2

PU2	$T_5$	$T_6$	$T_7$	$T_8$
Matlab (ms)	6	3	9	5

#### 1.2.2 Response time analysis for the CAN bus messages

Table 1.2: Response times for the CAN bus messages

CAN	$m_2$	$m_1$	$m_3$	$m_8$
Matlab (ms)	2	3	4	4

### 1.3 System model

### 1.4 Design decision

### 1.5 Results

Firstly: Response time analysis

Secondly: Control system input and output

# Chapter 2

## Part 2

### 2.1 Introduction

Analyzing and confirming broad analysis done on paper and in Matlab allows us to confirm the hypothesis made about the behavior of the CAN bus and its specific tasks. This analyses is in this case done in Inchron, which allows us to explore the real-time behavior of this embedded system in full detail. Some exploration on how the system should work has already been done in Part 1 by feeding the settings of the embedded system to the tool. The settings included is the hierarchy of the Processing Units (PUs) and their tasks which each have different periods, execution time and priority. The tree view, Figure 2.1, shows the details of the hierarchy from the Inchron's perspective.

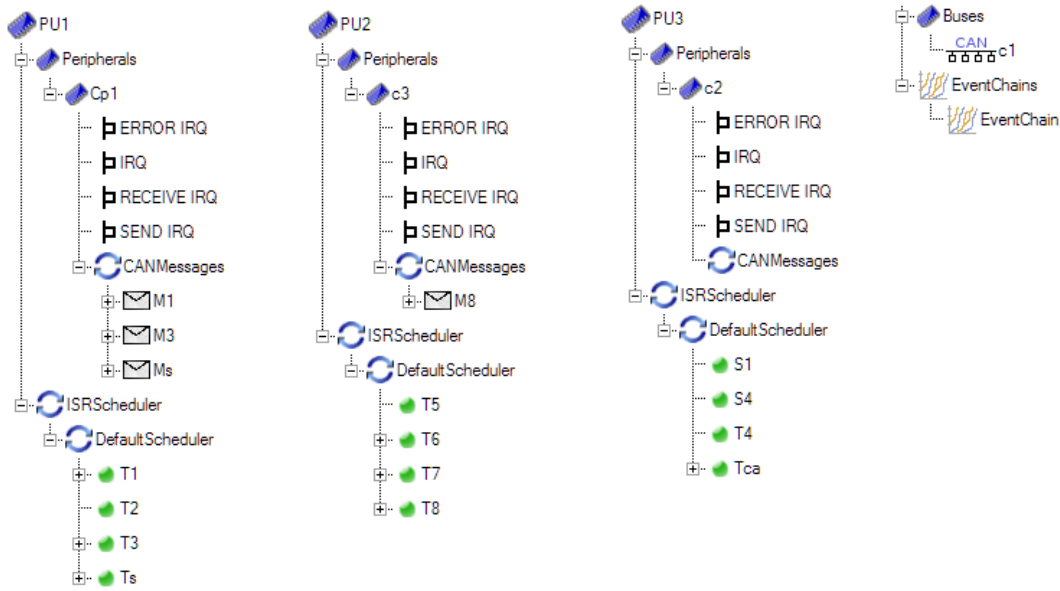


Figure 2.1: This shows the coarse grain hierarchy of the system ported into Inchron for verification and simulation of the real time components of our embedded system.

## 2.2 Response Time analysis

### 2.2.1 Response time analysis per processing unit

For this analysis all details have to be imported to Inchron as stated earlier. Now paying special attention to the messages which transfer the packets between controllers which make a full system.

When validating and simulating the model with the settings mentioned in Part 1 and Figure 2.1 and 2.4 we can observe the



Figure 2.2:



Figure 2.3:

Table 2.1: By running the Matlab script `ResponsetimeAnylysis.FPP.m` with the different parameters given for PU1 and PU2 these response times are obtained. These files are then delivered as PU1.m PU2.m

	PU1	$T_1$	$T_2$	$T_3$	$T_4 (T_s)$
Matlab (ms)	0.1	2.1	4.1	7.2	
Inchron (ms)	0.1	2.1	4.1	7.2	
	PU2	$T_5$	$T_6$	$T_7$	$T_8$
Matlab (ms)	6	3	9	5	
Inchron (ms)	6	3	9	5	

### 2.2.2 Response time analysis for the CAN bus messages

PU1 and PU2 are the only units within the system that are sending messages, shown for clarity in Figure 2.4, but PU3 contains the computing and actuating task which will receive the  $m_s$  message and mark the end of the sensor to actuator delay.

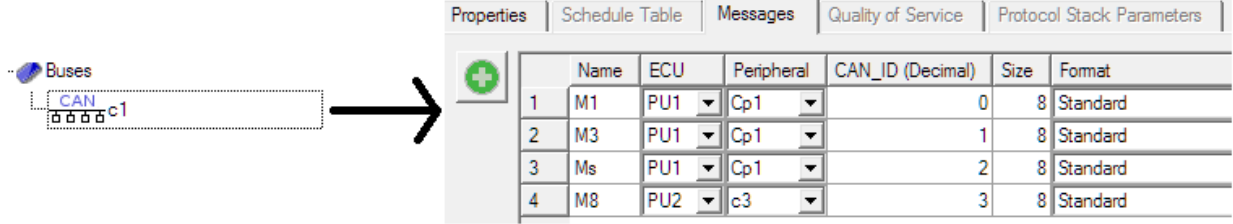


Figure 2.4: To

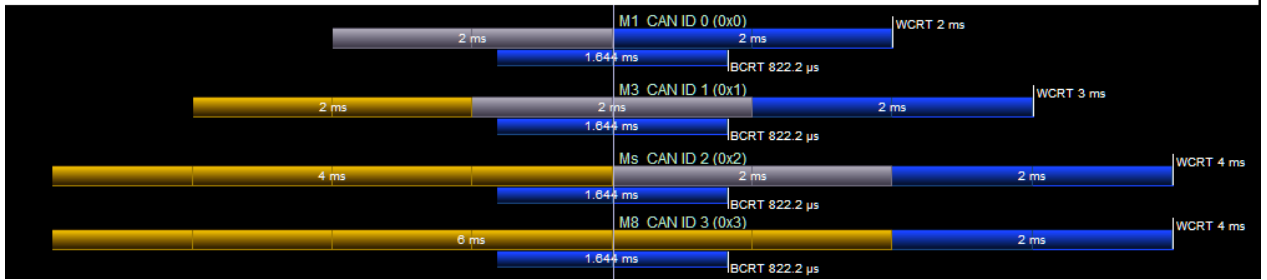


Figure 2.5: To

Table 2.2: Add caption

CAN	$m_2$	$m_1$	$m_3$	$m_8$
Matlab ms	2	3	4	4
Inchron	2	3	4	4

## 2.3 Optimisation for sensor-to-actuator delay

## 2.4 System model

## 2.5 Design decision

## 2.6 Results

Firstly: Response time analysis

Secondly: Plots from chronVIEW (before and after optimization)

Last: Control system input and output

## 2.7 Conclusions



# Chapter 3

## Part 3

### 3.1 Introduction

### 3.2 Answer all the questions

#### 3.2.1 Theoretical analysis versus actual implementation

### 3.3 Design decision

### 3.4 Results

Firstly: Solution to the design problem. (Include the parameters you have chosen)  
Secondly: from chronVIEW for your design

### 3.5 Conclusions

### 3.6 Results

### 3.7 Conclusion