Graphics Microsystems Inc.

PCU 100 Micro-controller

Project

Abstract:

This document contains all of the information regarding the operation of the PCU 100 (**P**ress **C**ontrol **U**nit).

Revision History

Author	Revision	Date	Remarks
Paul L Calinawan	00.01	November 26, 1996	Initial release
Paul L Calinawan	00.02	January 7, 1997	Made revisions based on the SPU Functional Specification Review
Date Printed : 12/17/2017			

Table of Contents

1. General Description....3
   1.1 Product Overview....3
   1.2 System Message Flow....4
2. Software....5
   2.1 Overview....5
   2.2 High Level Context Diagram....6
   2.3 General Protocol Format....8
   2.4 Servo Commands/Messages....9
      2.4.1 Command 0x41 - OK Status....9
      2.4.2 Command 0x42 - Set Unit Number....10
      2.4.3 Command 0x43 - Close Configuration Line....10
      2.4.4 Command 0x44 - Open Configuration Line....11
      2.4.5 Command 0x47 - Set Inter-Byte Delay....11
      2.4.6 Command 0x49 - Respond with D in Message....12
      2.4.7 Command 0x4A - Reset PCU System....12
      2.4.8 Command 0x4B - Set Timeout for A/D Settling....13
      2.4.9 Command 0x4E - Clear Status and Errors....14
      2.4.10 Command 0x51 - Set Coast Wait Time....14
   2.5 PCU Commands/Messages....15
      2.5.1 Command 0x52 - Set PCU Mode From SPU....15
      2.5.2 Command 0x53 - Move to Target A/D Position....16
      2.5.3 Command 0x54 - Get PCU Absolute Position....17
      2.5.4 Command 0x55 - Get Digital Input Status....18
      2.5.5 Command 0x56 - Set Digital Outputs....19
      2.5.6 Command 0x57 - Set A/D Fudge Factor....20
      2.5.7 Command 0x58 - Set Min A/D Position....21
      2.5.8 Command 0x59 - Set Max A/D Position....22
      2.5.9 Command 0x5A - Get Sheet Count....23
   2.6 Unsupported Commands....24
   2.7 Timing Specification....25
   2.8 Software UART Specification....25
   2.9 Servo Software Procedures....25
      2.9.1 Main Loop States....25
      2.9.2 Initialization State....26
      2.9.3 Baud Rate Configuration....27
      2.9.4 Hall Effect Interrupt Routine....28
      2.9.5 Motor Timing Specification....29
      2.9.6 Status LED Cadence Definition....40
      2.9.7 Timer Service....43
3. Hardware....44
   3.1 CPU....44
   3.2 Port Assignments....45
4. Issues....46

1. General Description

1.1 Product Overview

The SPU can operate in 2 distinct modes

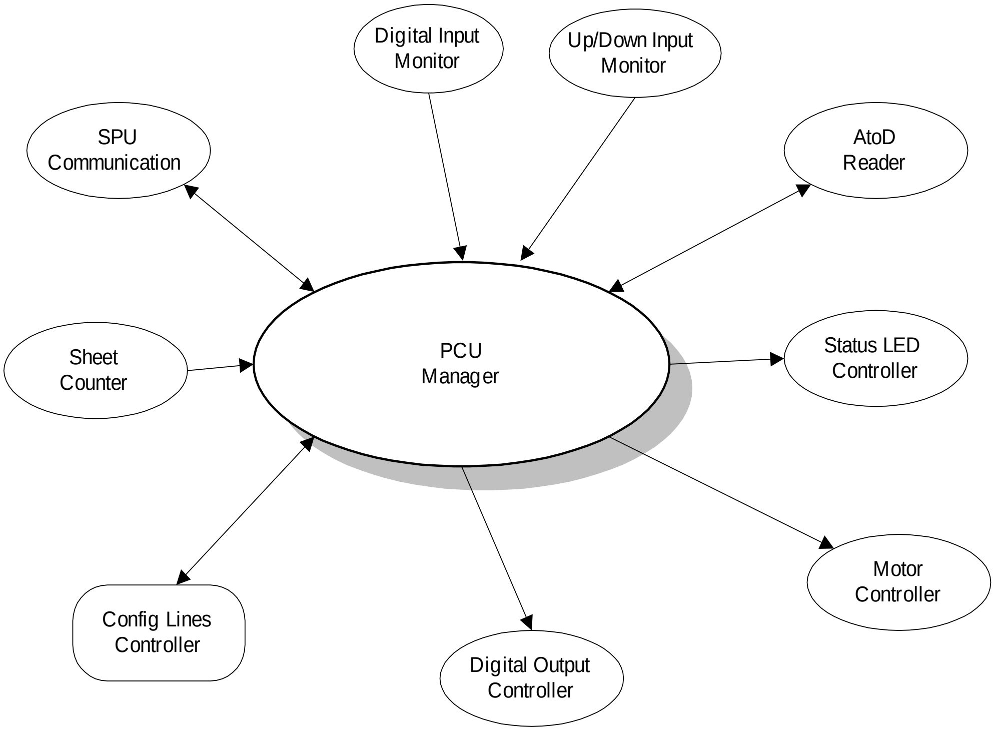
1.2 System Message Flow

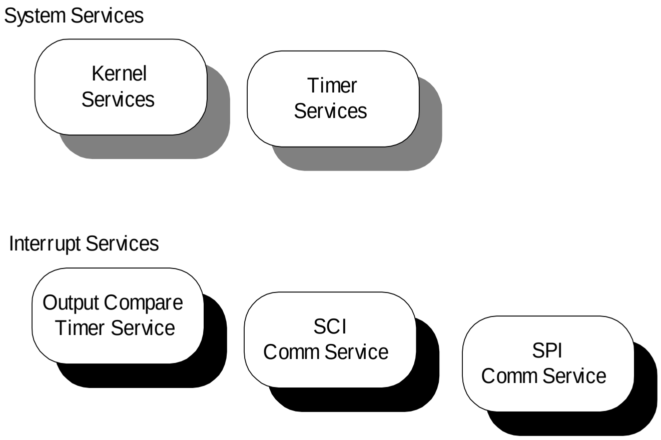
2. Software

2.1 Overview

2.2 High Level Context Diagram

The diagram below shows the different components involved in the operation of the PCU Micro-controller Unit.





Task Name	Task Description
PCU Manager	Performs the coordination of all the tasks within the system.
SPU Communicator	Is an interrupt driven task. This is in charge of all the communication tasks which includes reception and transmission of all messages on the SPU bus. Uses the 10 msec timing service to synchronize itself when the SPU bus is idle.
A to D Reader	Performs all the necessary actions to communicate with the A/D chip.
Motor Controller	With a given number of pulses (A to D equivalent) as well as the direction of the motor movement, the Motor Controller is in charge of applying pulses/power to the motor.
Digital Output Controller	Runs at every 60 msec. This task updates the state the of the 2 digital output ports.
Digital Input Monitor	Runs at every 30 msec. This task scans the 3 digital inputs to determine their states. One of the input's state is latched.
Config Line Controller	Normally active during initialization.
Status LED Controller	Running at every 60 msec, this updates the state the Status LED. The Status LED provides several different cadences to indicate several different states of the PCU module.
Timer Services	Provides a basic timing service of 10 msec or the "System Tick". Using this as a reference, two other timers are available : 30 msec timer 60 msec timer
IRQ Isr	Is in charge of servicing the interrupts generated by the presence of data on the SPU communications bus. Each data byte is processed and queued for the SPU Communicator Task.

2.3 General Protocol Format

2.4 Servo Commands/Messages

FULL This term is used to indicate that a given message is . .. .he SPU uses this exact same message for the Old Servo Micro-controller as well as the PCU 100 module. The SPU however needs to know what kind of

2.4.1 Command 0x41 - OK Status

From SPU :

Cmd
0x41

PCU Id
0 - 0xFF

Coming from the SPU unit : This is considered a special configuration command and is normally followed by the PCU Id which sets the initial ID of a given unit.

Another form of this command is in a full packet format :

Len	PCU Id	Cmd	CRC 1	CRC 2
0x05	0xID	0x41	CRC 1	CRC 2

In this format the SPU is requesting to check the presence of a given Servo Unit. The Servo Unit with the matching ID replies with a 1 Byte message as shown below.

From Servo Unit :

Cmd
0x41

Compatibility: **FULL**

Description :
ASCII “A”

Coming from the servo unit : This is a special 1 Byte message
(No Header, CRC etc.). The Servo Unit simply returns a 0x41 to the SPU indicating its presence within the string.

2.4.2 Command 0x42 - Set Unit Number

From SPU :

Len	PCU Id	Cmd	Data 1	Data 2	CRC 1	CRC 2
0x07	Old ID	0x42	New ID	XxXX	CRC1	CRC2

Note : XxXX - Signifies Don't Care .

Compatibility: **FULL**

Description :
ASCII "B"

This command allows the SPU to give a new PCU Id to a Servo Unit that has already been configured. The initial Unit Number (ID) is configured during the initialization or configuration phase of the Servo Units.

2.4.3 Command 0x43 - Close Configuration Line

From SPU :

Len	PCU Id	Cmd	CRC 1	CRC 2
0x05	0xID	0x43	CRC 1	CRC 2

Compatibility: **FULL**

Description :
ASCII "C"

This command gives the SPU the ability to set the ConfigLine of a given servo unit so that the other units within the string can start to communicate. A closed line allows the next servo within the string to communicate.

Note : The servo units themselves control the ConfigLine most of the time.

2.4.4 Command 0x44 - Open Configuration Line

From SPU :

Len	PCU Id	Cmd	CRC 1	CRC 2
0x07	0xID	0x44	CRC 1	CRC 2

Compatibility: **FULL**

Description :
ASCII “D”

This command gives the SPU the ability to clear the ConfigLine of a given servo unit so that the other units within the string can start to communicate. An open line indicates that the next servo within the string can not communicate.

2.4.5 Command 0x47 - Set Inter-Byte Delay

This command was originally called “Set Transmission Delay”.

From SPU :

Len	PCU Id	Cmd	Data 1	Data 2	CRC 1	CRC 2
0x07	0xID	0x47	**XDLY**	XxXX	CRC 1	CRC 2

Description :
ASCII “G”

This is a command that sets the inter-byte delay.

XDLY

Default Value :

0x01 =

We need to count CPU cycles.

Each count is X msec

2.4.6 Command 0x49 - Respond with D in Message

From SPU :

Len	PCU Id	Cmd	CRC 1	CRC 2
0x05	0xID	0x49	CRC 1	CRC 2

Description :
ASCII “I”

This is actually a request message from the SPU asking the Servo Units their type. The Old Servo Micro-controller would send a “C” or 0x43. The PCU 100 will send a “D” or 0x44. This is necessary in order for the SPU to differentiate the type of devices that are connected within the string..

From Servo Unit :

Cmd
0x44

This is a special 1 Byte reply. These Servo Units respond with an ASCII - “D” (0x44) which denotes a PCU 100 type servo.

2.4.7 Command 0x4A - Reset PCU System

From SPU :

Len	PCU Id	Cmd	CRC 1	CRC 2
0x05	0xID	0x4A	CRC 1	CRC 2

Description :
ASCII “J”

This command causes the Servo to re-initialize the software. The software executes a full initialization routine starting from the very top of the main loop.

2.4.8 Command 0x4B - Set Timeout for A/D Settling

From SPU :

Len	PCU Id	Cmd	Data 1	Data 2	CRC 1	CRC 2
0x07	0xID	0x4B	**TPLS**	XxXX	CRC 1	CRC 2

Description :
ASCII “K”

This command was originally called “Set Timeout for Pulses”. This command sets the timeout parameter used for determining if the motor is still moving through the use of the AtoD reading.

TPLS

Default Value :

Each count is X msec

2.4.9 Command 0x4E - Clear Status and Errors

From SPU :

Len	PCU Id	Cmd	CRC 1	CRC 2
0x05	0xID	0x4E	CRC 1	CRC 2

Description :
ASCII “N”

This command simply allows the SPU to clear all errors latched by the Status Byte. This message is generally by the SPU after it detects an error from a message reply of a Servo Unit.

The Digital Input Latch Register is also cleared. See

Status Byte Definition :

2.4.10 Command 0x51 - Set Coast Wait Time

From SPU

Len	PCU Id	Cmd	Data 1	Data 2	CRC 1	CRC 2
0x07	0xID	0x51	Coast Time	XxXX	CRC 1	CRC 2

Description :
ASCII “Q”

This command allows the setting of the coast wait time parameter.

Default 0x20 = 100 msec

2.5 PCU Commands/Messages

The following sections describes the new messages that were specifically designed for the PCU module.

It was intentional to designate new command numbers for the PCU instead of reusing the commands from the old servo. This will be useful for debugging as well.

2.5.1 Command 0x52 - Set PCU Mode

From SPU

Description :
ASCII “ ”

Len	PCU Id	Cmd	Data 1	Data 2	CRC 1	CRC 2
0x07	0xID	0x53	**PMOD**	XxXX	CRC 1	CRC 2

2.5.2 Command 0x53 - Move to Target A/D Position

From SPU

Description :
ASCII “ ”

Len	PCU Id	Cmd	Data 1	Data 2	CRC 1	CRC 2
0x07	0xID	0x53	**TPOS1**	**TPOS2**	CRC 1	CRC 2

TPOS2 contains the Least Significant Byte while TPOS1 contains the Most Significant Byte. Both TPOS1 and TPOS2 form the 12 bit value of the A/D reading.

Below is a table showing an expanded view of TPOS1 and TPOS2.

Areas that are shaded indicate that it is unused and are masked by the PCU.

<table>
<tr><th colspan="8">TPOS1</th><th colspan="8">TPOS2</th></tr>
<tr><td>▨</td><td>▨</td><td>▨</td><td>▨</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">mask 0x7F</td><td colspan="8">mask 0xFF</td></tr>
</table>

2.5.3 Command 0x54 - Get PCU Absolute Position

From SPU

Len	PCU Id	Cmd	CRC 1	CRC 2
0x05	0xID	0x54	CRC 1	CRC 2

Description :
ASCII “”

This command requests the PCU to send back its absolute 12-bit A/D reading. This is considered the current the position. This position reading may change if to the “press operator” decides to change the position manually. This command allows the SPU to react accordingly to any positional changes made by the operator.

From PCU

Len	PCU Id	Stat	Data 1	Data 2	CRC 1	CRC 2
0x07	0xID	ERR STAT	**D1**	**D2**	CRC 1	CRC 2

D2 contains the Least Significant Byte while D1 contains the Most Significant Byte.

Both D1 and D2 form the 12 bit value of the A/D reading.

Below is a table showing an expanded view of D1 and D2.

Areas that are shaded indicate that it is unused.

D1								**D2**							
15	▨	▨	▨	11	10	9	8	7	6	5	4	3	2	1	0

Bit 15 Contains the Direction of last move.

0 = forward
1 = reverse

2.5.4 Command 0x55 - Get Digital Input Status

The PCU provides 3 digital inputs.

From SPU

Len	PCU Id	Cmd	CRC 1	CRC 2
0x05	0xID	0x55	CRC 1	CRC 2

Description :
ASCII “T”

This command requests the PCU to send back the status of it’s 3 input ports.

From PCU

Len	PCU Id	Status	Data 1	Data 2	CRC 1	CRC 2
0x07	0xID	**ERR STAT**	**INPUT DATA**	XxXX	CRC 1	CRC 2

Below is a table showing an expanded view of the INPUT DATA

INPUT DATA							
7	6	5	4	3	2	1	0
High Nibble (Latched State)				**Low Nibble (Current State)**			
▨	Inp3	Inp2	Inp1	▨	Inp3	Inp2	Inp1

Whenever a current state is detected to be set, that value is latched.

The SPU has to send a command to clear the latch values

2.5.5 Command 0x56 - Set Digital Outputs

From SPU

Len	PCU Id	Cmd	Data 1	Data 2	CRC 1	CRC 2
0x07	0xID	0x56	**OUT DATA**	XxXX	CRC 1	CRC 2

Description :
ASCII “U”

This command requests the PCU to activate or deactivate any of its 2 digital outputs.

The table below shows which bits control which Digital Output ports:

OUT DATA							
7	6	5	4	3	2	1	0
						Output 2	Output 1

Bit Val	**Output State**
1	Activated
0	Deactivate

2.5.6 Command 0x57 - Set A/D Fudge Factor

From SPU

Len	PCU Id	Cmd	Data 1	Data 2	CRC 1	CRC 2
0x07	0xID	0x57	**FUDGE**	XxXX	CRC 1	CRC 2

Description :
ASCII “ ”

This command allows the SPU to adjust the fudge factor of the A/D readings.

2.5.7 Command 0x58 - Set Min A/D Position

From SPU

Len	PCU Id	Cmd	Data 1	Data 2	CRC 1	CRC 2
0x07	0xID	0x58	**MIN POS1**	**MIN POS2**	CRC 1	CRC 2

Description :
ASCII “ ”

The PCU has the capability of being controlled manually by the operator. This command allows the SPU to adjust the maximum position that the PCU will allow the servo to move.

MIN POS2 contains the Least Significant Byte while **MIN POS1** contains the Most Significant Byte. Both **MIN POS1** and **MIN POS2** form the 12 bit value of the A/D reading.

Below is a table showing an expanded view of **MIN POS1** and **MIN POS2**.

Areas that are shaded indicate that it is unused and are masked by the PCU.

MIN POS1								**MIN POS2**							
▨	▨	▨	▨	11	10	9	8	7	6	5	4	3	2	1	0
mask 0x7F								mask 0xFF							

2.5.8 Command 0x59 - Set Max A/D Position

From SPU

Len	PCU Id	Cmd	Data 1	Data 2	CRC 1	CRC 2
0x07	0xID	0x58	**MAX POS1**	**MAX POS2**	CRC 1	CRC 2

Description :
ASCII “ ”

The PCU has the capability of being controlled manually by the operator. This command allows the SPU to adjust the maximum position that the PCU will allow the servo to move.

MAX POS2 contains the Least Significant Byte while **MAX POS1** contains the Most Significant Byte. Both **MAX POS1** and **MAX POS2** form the 12 bit value of the A/D reading.

Below is a table showing an expanded view of **MAX POS1** and **MAX POS2**.

Areas that are shaded indicate that it is unused and are masked by the PCU.

MAX POS1								**MAX POS2**							
▨	▨	▨	▨	11	10	9	8	7	6	5	4	3	2	1	0
mask 0x7F								mask 0xFF							

2.5.9 Command 0x5A - Get Sheet Count

Sheet counters that provides 25 sheets per/second has a cycle of 40 milliseconds. This command needs to be executed every xxxx seconds to avoid overflow. The SPU is incharge of clearing the counter. You have to clear it within 255 counts after getting the sheet count.

From SPU

Len	PCU Id	Cmd	CRC 1	CRC 2
0x05	0xID	0x5A	CRC 1	CRC 2

Description :
ASCII “T”

This command requests the PCU to send back the current number of pulses logged by the sheet counter of the PCU.

From PCU

Len	PCU Id	Status	Data 1	Data 2	CRC 1	CRC 2
0x07	0xID	**ERR STAT**	**SC1**	**SC2**	CRC 1	CRC 2

SC1 and SC2 form the 16 Bits of sheet count data.

SC2 contains the Least Significant Byte.
SC1 contains the Most Significant Byte.

The SPU should get the sheet count and clear the counter.

The Clear status command normally is executed after the SPU is able to retrieve the sheet count correctly.

Due to the maximum number that the sheet counter can log, the SPU has to retrieve this information before an overflow occurs.

Calculation : Maximum wait :

2.6 Unsupported Commands

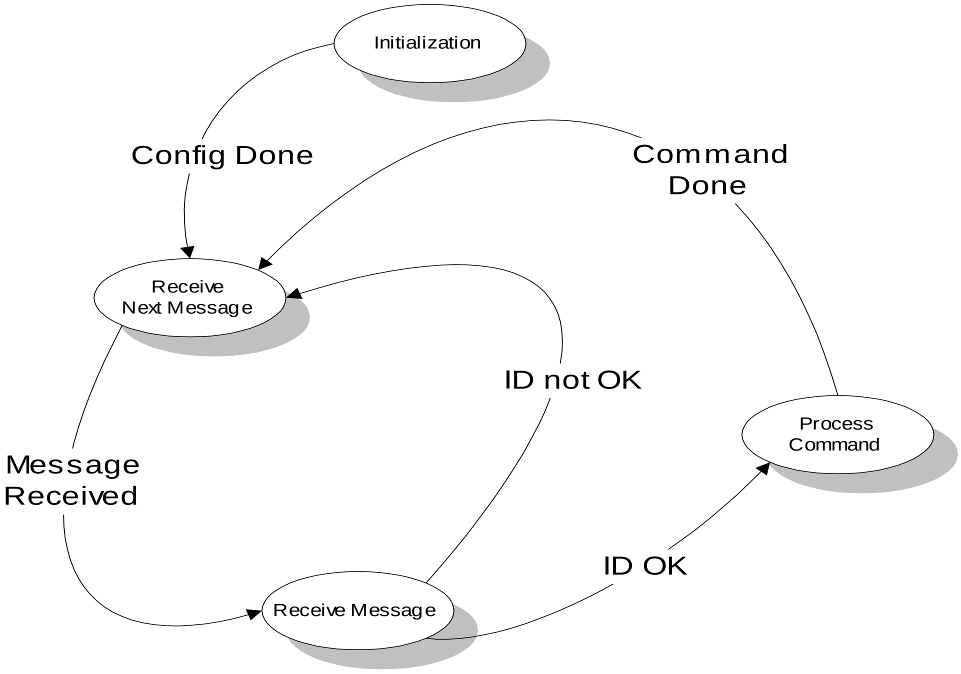
2.7 Timing Specification

2.8 Software UART Specification

2.9 Servo Software Procedures

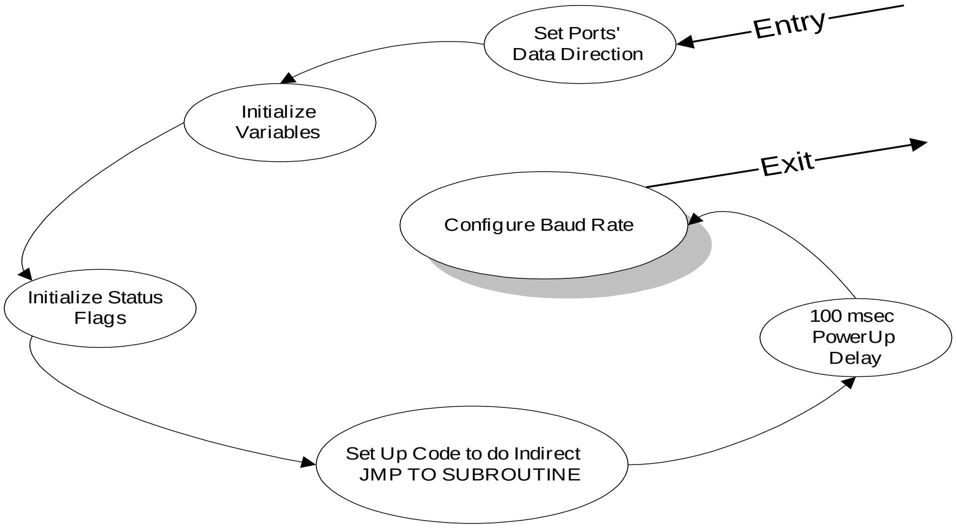
2.9.1 Main Loop States

Procedures with shadows are exploded to present more details.
The diagram below shows the main executive procedures that the old servo performs :



2.9.2 Initialization State

The diagram below shows the initialization procedure :



2.9.3 Baud Rate Configuration

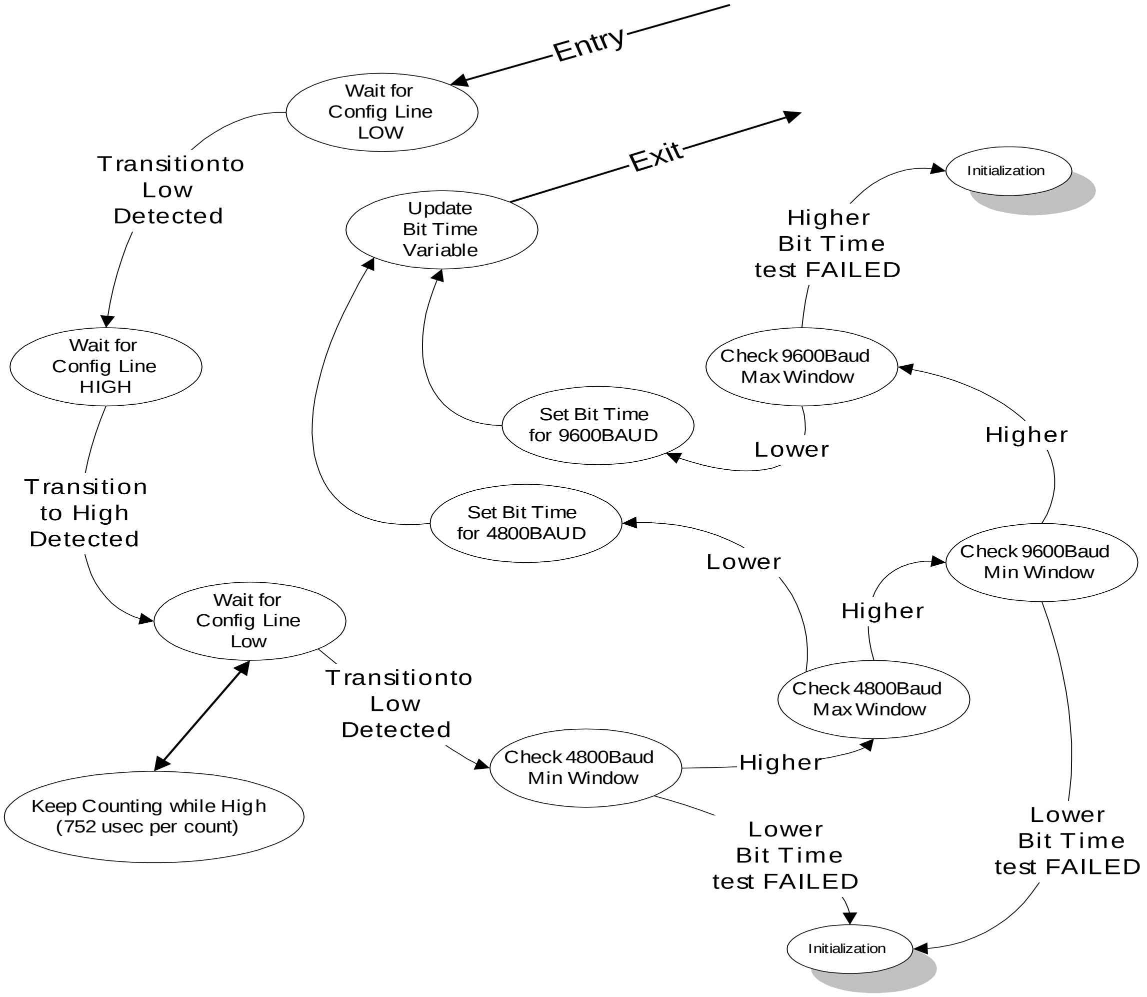
This procedure measures the time that the Config Line stays high and uses it as a reference to produce the desired communications speed. This type of speed calibration was implemented for 2 reasons

1. Some servos within the string can only communicate at 4600 BAUD.
2. Due to price sensitivity, the servos do not use a crystal for operation.

The SPU sends the following pulse:

48 msec = 4800BAUD
22 msec = 9600BAUD

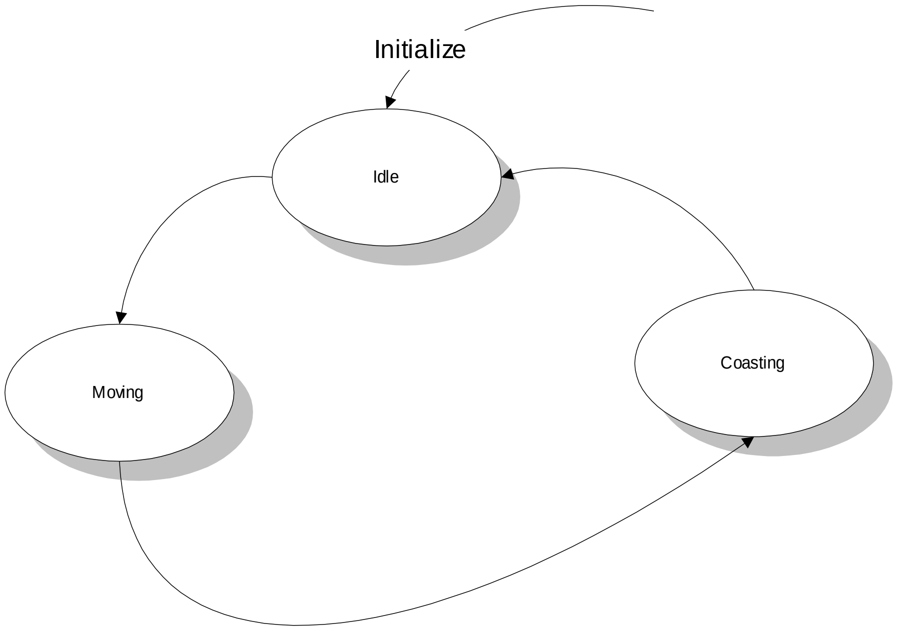
The diagram below shows the Baud Rate Configuration procedure :


Entry
Wait for Config Line LOW
Exit
Transitionto Low Detected
Update Bit Time Variable
Initialization
Higher Bit Time test FAILED
Wait for Config Line HIGH
Check 9600Baud Max Window
Set Bit Time for 9600BAUD
Lower
Higher
Transition to High Detected
Set Bit Time for 4800BAUD
Lower
Check 9600Baud Min Window
Higher
Wait for Config Line Low
Transitionto Low Detected
Check 4800Baud Max Window
Check 4800Baud Min Window
Higher
Keep Counting while High (752 usec per count)
Lower Bit Time test FAILED
Lower Bit Time test FAILED
Initialization

2.9.4 Hall Effect Interrupt Routine

2.9.5 Motor Control States

These are the 3 states of servo the motor :



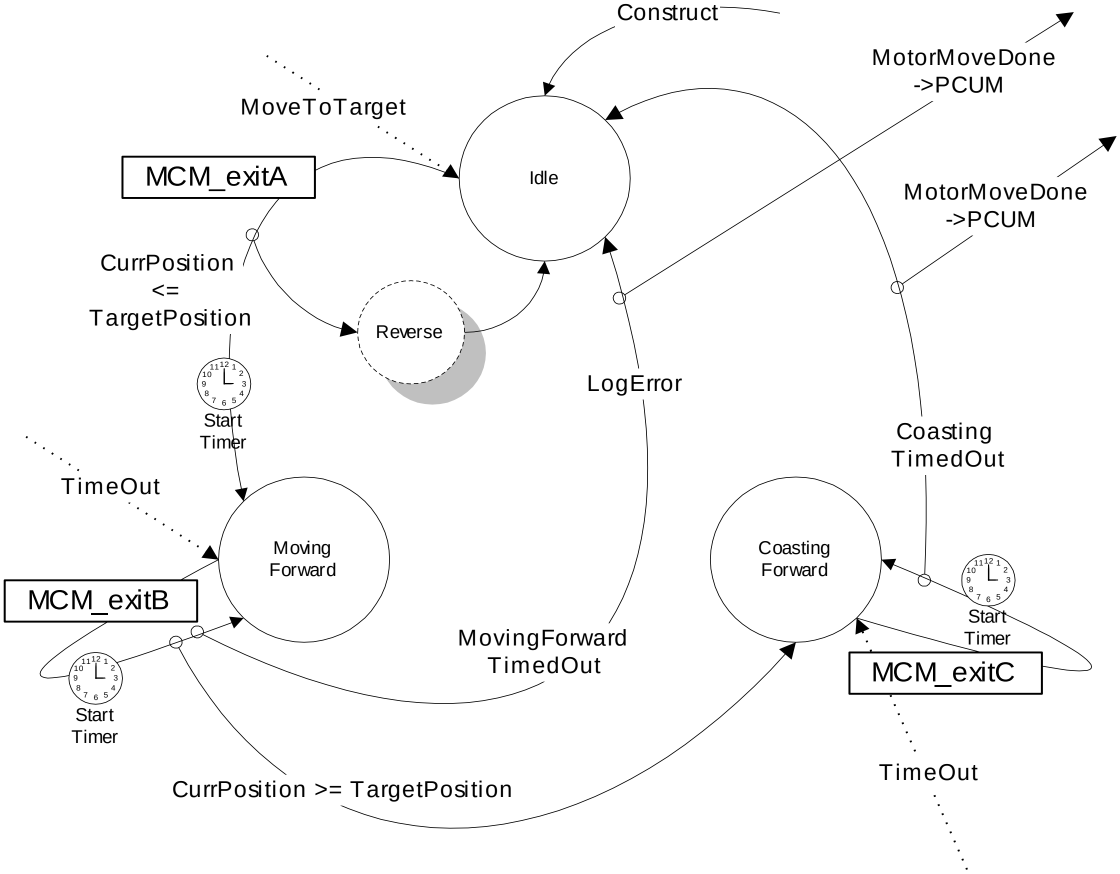
Upon power up, the motors are placed in the idle state.

When a move command is received, the motors are initiated to move in a given direction This is the Moving State. This state is further shown in detail in the next pages.

When the servo has reached its target position (pulse count), it then goes in the Coasting State. This state is also shown in detail in the next pages.

2.9.6 Moving and Coasting Motor States

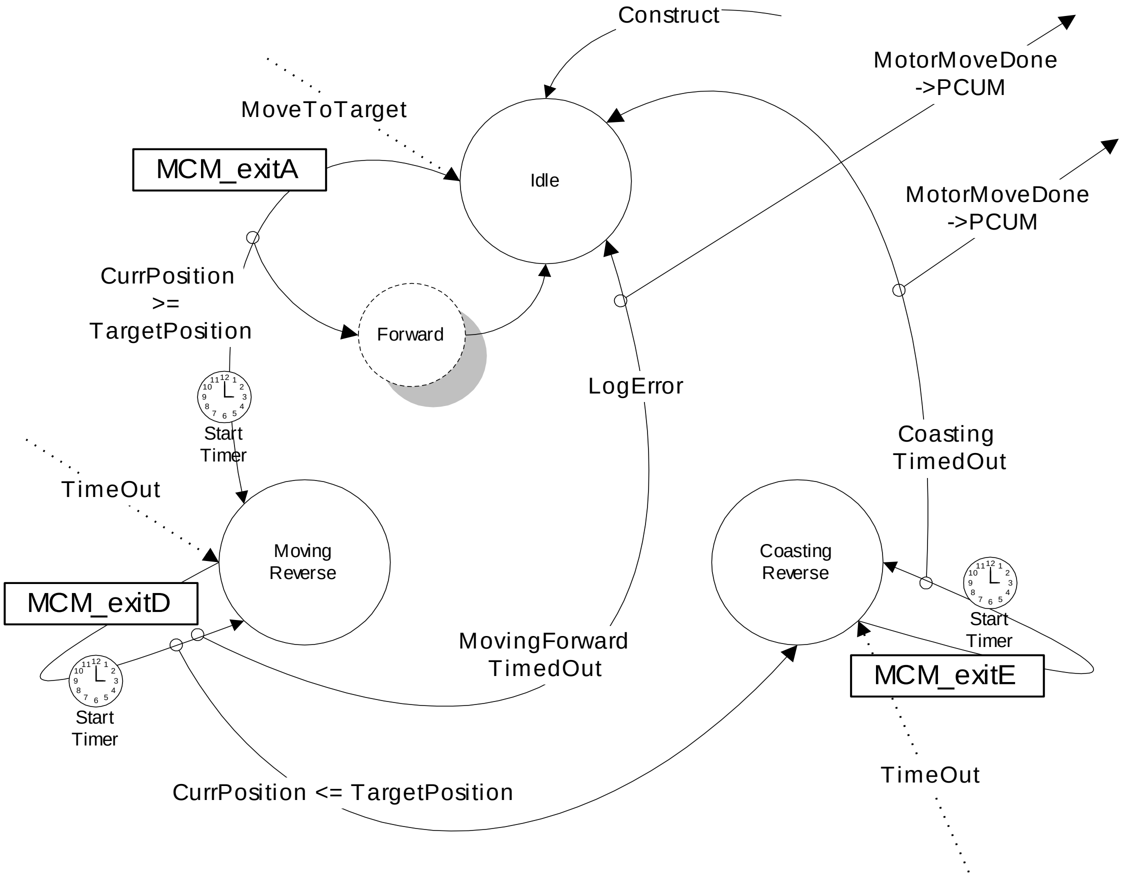
2.9.6.1.1 Forward Moving and Coasting States



Virtual Message:

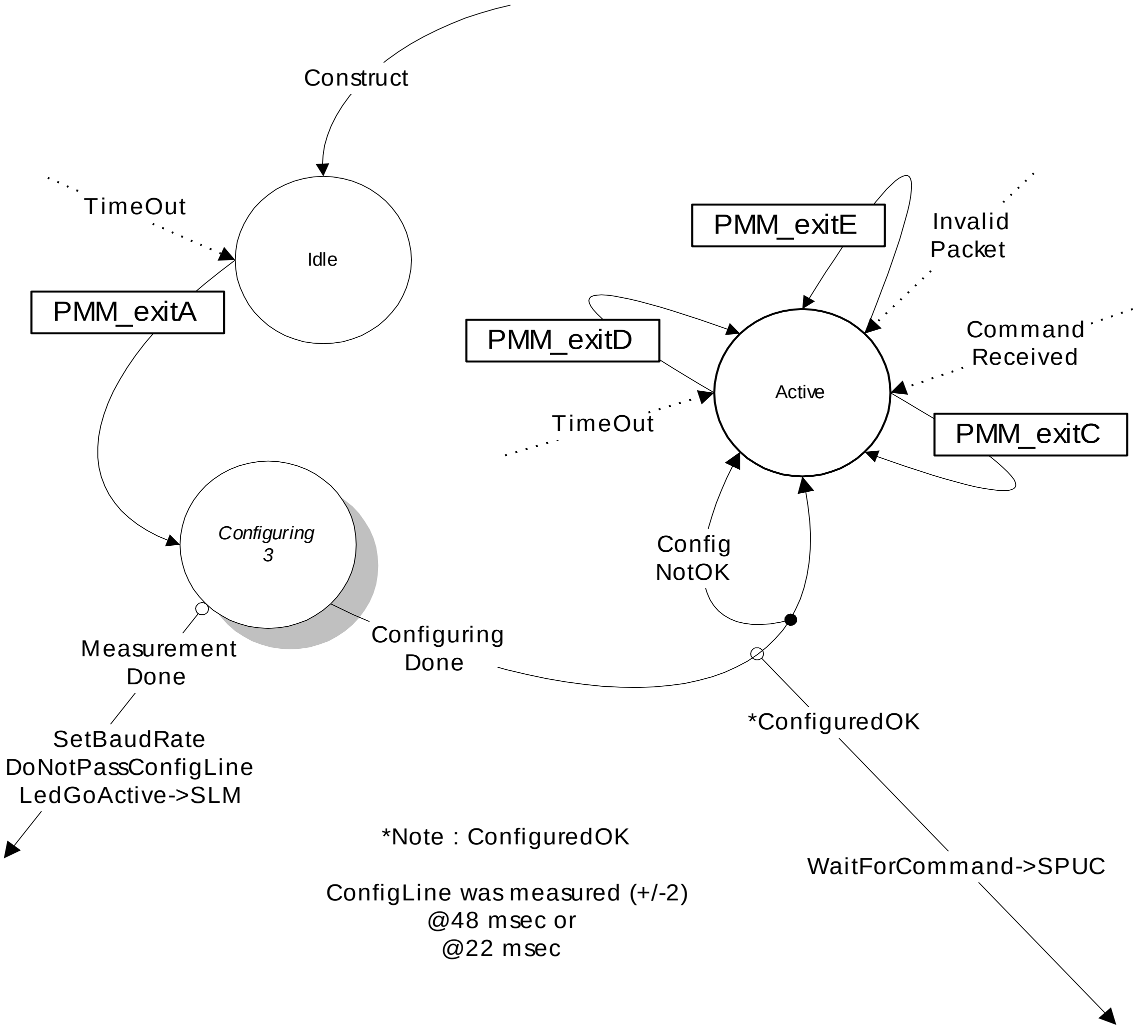
SetTarget(newTarget)
SetMovingTimeOut(newMovingTimeOut)
SetCoasting(newCoastingTimeOut)

2.9.6.1.2 Reverse Moving and Coasting States



PCU Manager Machine (PMM)

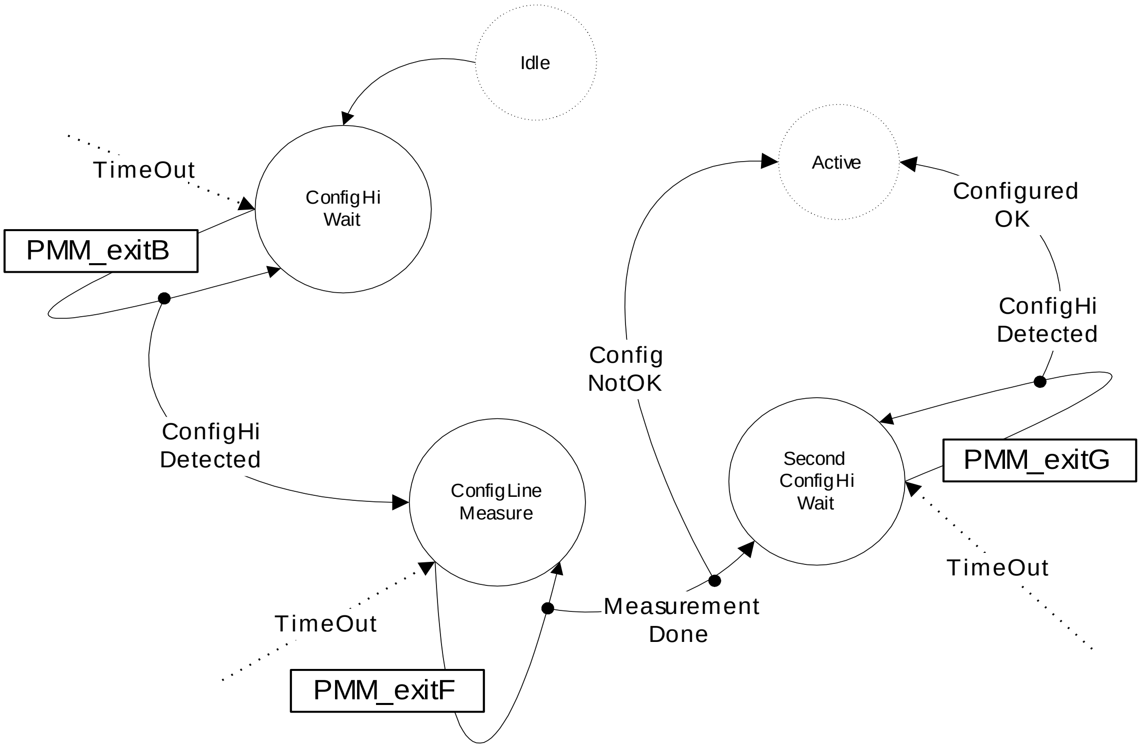
2.9.6.2 PMM Operation



*Note : ConfiguredOK

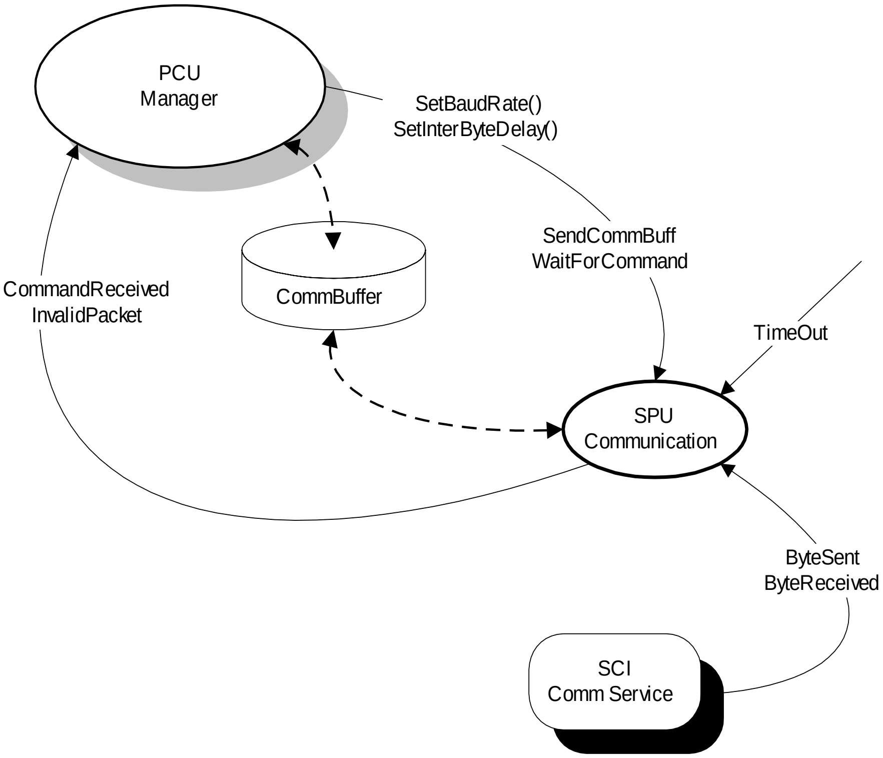
ConfigLine was measured (+/-2)
@48 msec or
@22 msec

2.9.6.2.1 PMM Operation, Configuring

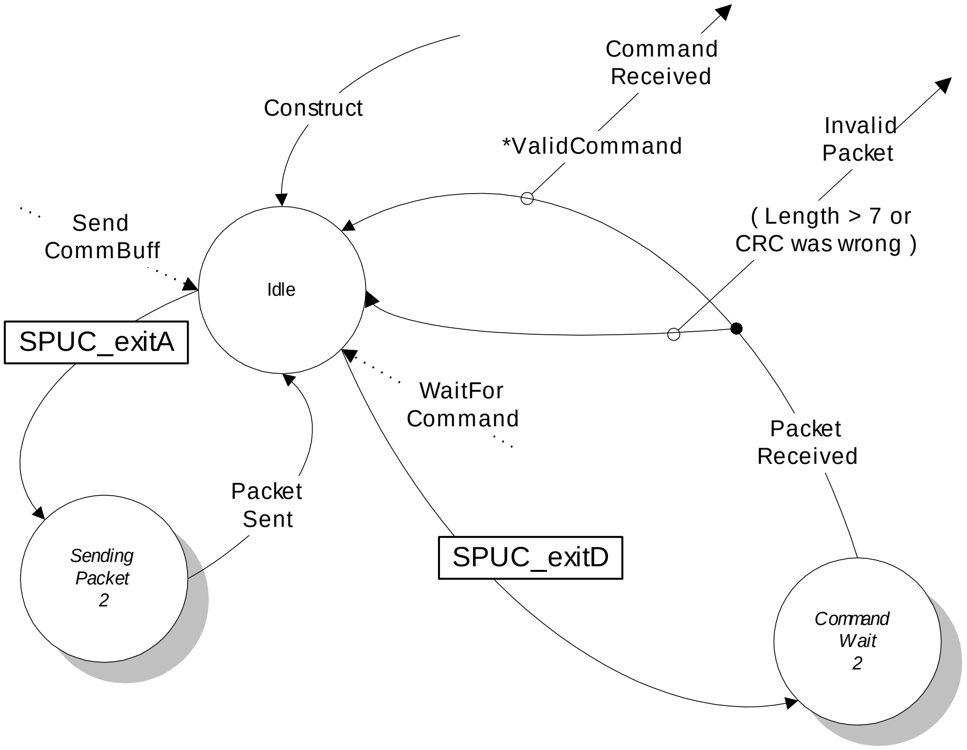


2.9.7 SPU Communication Machine (SPUC)

2.9.7.1 SPUC System Interaction



2.9.7.2 SPUC Operation



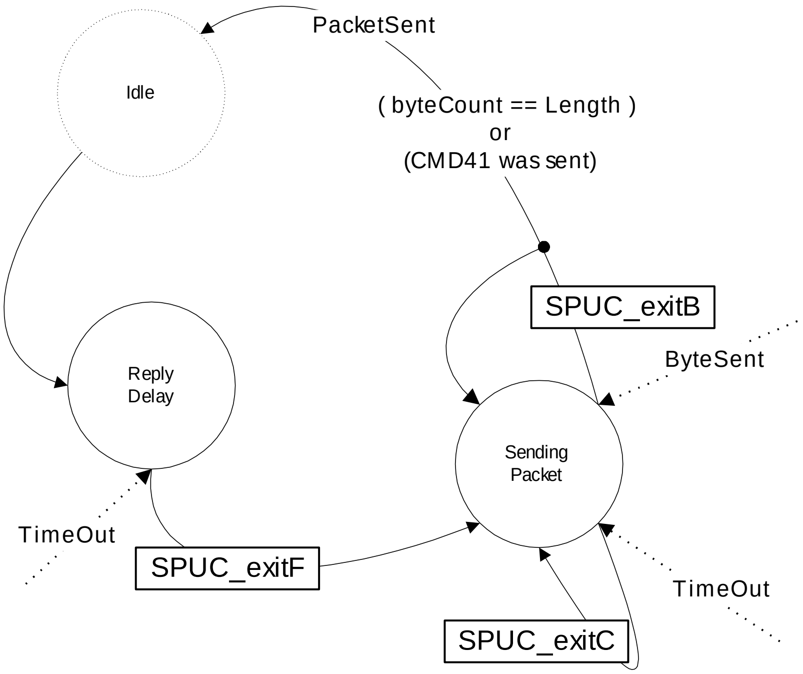
*Note : ValidCommand

(Length == byteCount && ID mathces)
or
(CMD 41+ID was received)

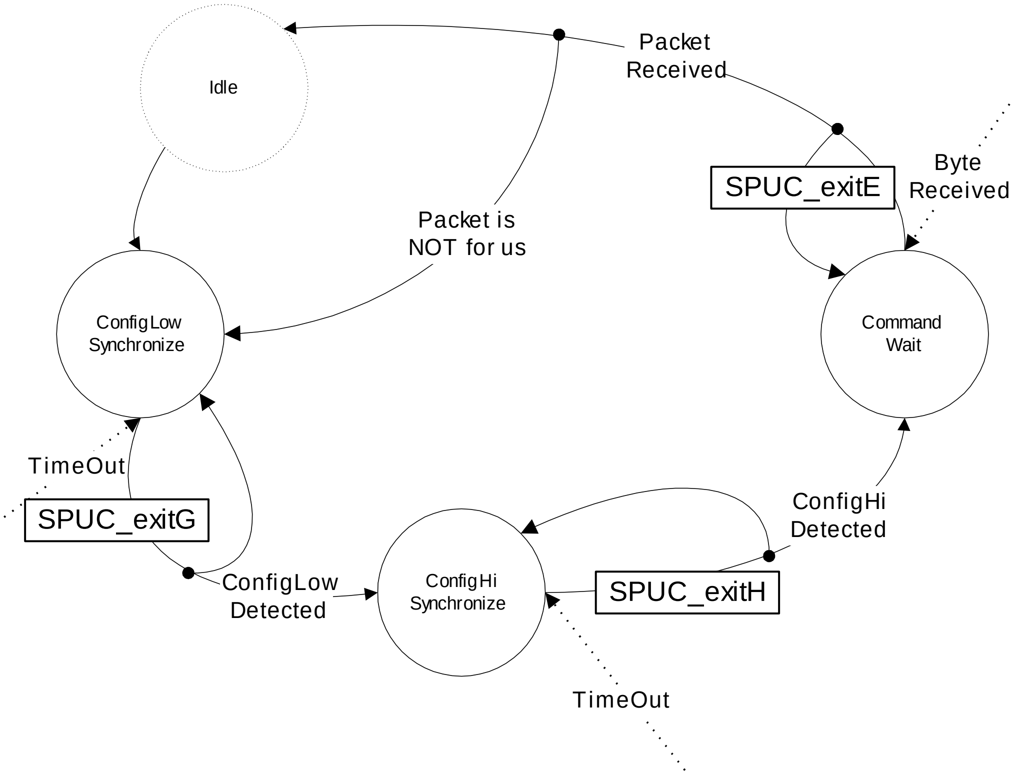
Virtual Message:

SetBaudRate(baudRate)
SetInterByteDelay(ibyteDelay)
SetReplyDelay(rplyDelay)

2.9.7.2.1 SPUC Operation, Sending Packet



2.9.7.2.2 SPUC Operation, Command Wait



2.9.8 Status LED Cadence Definition

A cadence will complete at least one cycle (16 states) before it can be changed.

□	LED off
▨	LED on

Scale : 60 msec

Not Powered

□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

STEADY : Powered / Not Initialized

▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

SLOW PULSE : Powered / Initialized

▨	▨	▨	▨	▨	▨	▨	▨	□	□	□	□	□	□	□	□

FAST PULSE : Powered / Sending Message to SPU

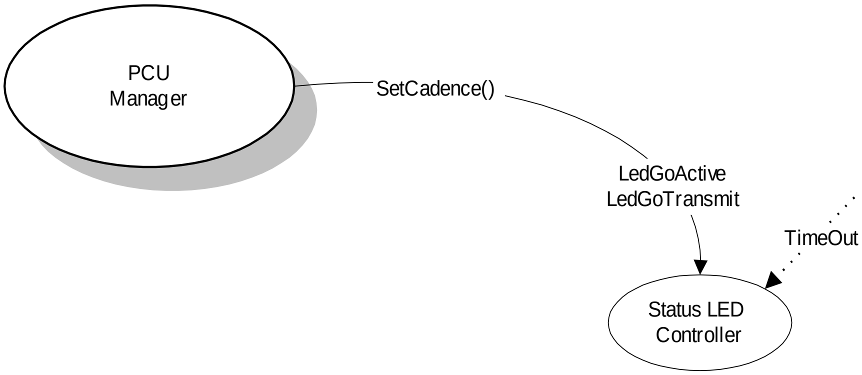
▨	□	▨	□	▨	□	▨	□	▨	□	▨	□	▨	□	▨	□

STROBE : Powered / Error Detected

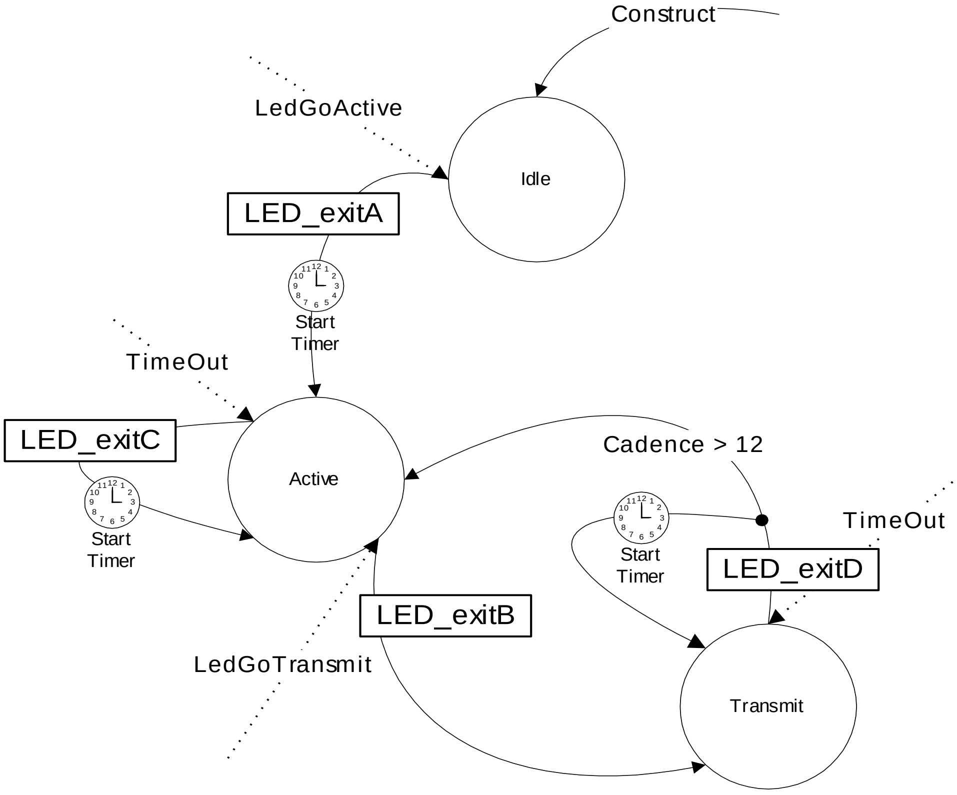
▨	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

2.9.9 Status LED Machine (SLM)

2.9.9.1 SLM System Interaction



2.9.9.2 SLM Operation



Virtual Message:

SetCadence(newCadence)

2.9.10 Timer Service

The Output Compare (OC) function on the 6805P9 is used to provide a 4 millisecond basic timing service. An Interrupt Service Routine is in charge of updating the timer tick count and restarting the OC timer.

3. Hardware

3.1 CPU

3.2 Port Assignments

The table below shows how each of the ports of the Motorola HC6805 P9 are used.

Pin Number	Port Name	Usage
1	**RESET Low**	**To Reset Circuit**
2	**IRQ Low**	**Comm Input, Interrupt source**
3	**PA7**	**Comm Input, data input from SPU**
4	**PA6**	**Comm Output, data output to SPU**
5	**PA5**	**Xmit On (High), Xmit Off (Low)**
6	**PA4**	**Config PASS**
7	**PA3**	**Config IN**
8	**PA2**	**Motor Control (Direction : Forward)**
9	**PA1**	**Motor Control (Direction : Reverse)**
10	**PA0**	unused / available
		SIOP function is unused
11	SD0 **(PB5)**	**CHIP SEL for A to D communication**
12	SDI (**PB6)**	**A to D communication data input port**
13	SCK (**PB7)**	**A to D communication clock source**
14	**Vss**	**CPU Power Ground**
		Note : *PC3-PC7 A/D function is unused.*
15	**PC7** (Vrh)	**PCU Servo UP**
16	**PC6** (AN0)	**PCU Servo DOWN**
17	**PC5** (AN1)	**PCU Digital Input 3**
18	**PC4** (AN2)	**PCU Digital Input 2**
19	**PC3** (AN3)	**PCU Digital Input 1**
20	**PC2**	**PCU Digital Output 2**
21	**PC1**	**PCU Digital Output 1**
22	**PC0**	**Status LED Port**
23	**PD5**	unused / available
24	TCMP	unused (Output Compare Port)
25	**PD7** (TCAP)	**Sheet Counter Input**
26	**OSC2**	4.00 Mhz crystal oscillator
27	**OSC1**	4.00 Mhz crystal oscillator
28	**Vdd**	CPU +5V Power Source

4. Issues