# Psuedo Code for UART

//Initialize UART

bool InitializeUART ( uint8\_t Priority )

{

PC4 RX input

PC5 TX output

Enable the UART module using the RCGCUART register

Wait for the UART to be ready (PRUART)

Enable the clock to the appropriate GPIO module via the RCGCGPIO, PortC

Wait for the GPIO module to be ready (PRGPIO)

Configure the GPIO pins

Select the Alternate function for the UART pins

Configure the PMCn fields in the GPIOPCTL register to assign the UART pins

Disable the UART by clearing the UARTEN bit in the UARTCTL register.

Write the integer portion of the BRD to the UARTIBRD register (baud rate = 9600)

Write the fractional portion of the BRD to the UARTFBRD register.

Write the desired serial parameters to the UARTLCRH register.

Configure the UART operation using the UARTCTL register.

Enable the UART by setting the UARTEN bit in the UARTCTL register.

Interrupt, UART4: interrupt 60

ReceiveState = ES\_WaitFor7E;

ES\_Timer\_InitTimer(UartTimer, 1000);

Pairing array initialization

Initialize the sending data array

Starter

Data Length MSB

Data length LSB

API ID

Frame ID

Destination Address MSB

Destination Address LSB

Options

Actual Data Header

Control Array initialization

Control data intialization

Starter

Data Length MSB

Data Length LSB

API ID

Frame ID

Destination Address MSB

Destination Address LSB

Options

Actual Data Header

Pair or Unpair

return true;

}

Service function for the states of the game

ES\_Event RunUARTSM( ES\_Event ThisEvent )

{

//States for the game

If the current state the card slide

If the current event received is ES\_CARDSLIDE

Change state to State\_UNPAIRED

Else if the current state the State\_UNPAIRED

If the current event received is EV\_REQUESTPAIR

Calculate the checksum and move to the next state State\_REQUEST2PAIR

Init timer for the timemout for pairing

Send the first byte for pair

Else if the current state the State\_REQUEST2PAIR

If the event received is EV\_ACKRECEIVED

Change state to State\_PAIRED

If the timeout for pairing even is received then move back to unpaired state

Else if the current state the State\_PAIRED

If the 200ms timer is timeout

Query the button states and group the packet and send out

If the lost connection timer is timemout

Go back to the unpaired state and reinitialized the variables and counters.

return ReturnEvent;

}

//Service function for interrupt

void UARTInterrupt()

{

//Transmit interrupt

If the interrupt is determined to be transmit

If the current state is unpair

Transmit the data for paring

Else if the current state is paired

Transmit the data for control

//Receiving interrupt

If the interrupt is determined to be receiving

//Ready to pair state

If the current state is State\_REQUEST2PAIR

Suck up the result packet or suck up the incoming data packet

If the current state is State\_PAIRED

Suck up the result packet or suck up the incoming data paket

If encounter other group’s broadcasting data packet, ignore it.

}

# Psuedo Code for 7 segment

Create table for the 7 segment hardware

Initialize the 7 segment I/O ports

Query the state of the potentiometer and use pointer to point to the corresponding 7 segment table

Display the 7 segment using blocking code

# Psuedo Code for Button Debounce

We have 4 buttons, the debounce code is the same. Here is the structure.

bool InitializeButton ( uint8\_t Priority )

{

Initialize the button I/O ports

Init timer for button debounce

return true;

}

ES\_Event RunButtonDebounceSM( ES\_Event ThisEvent )

{

If button state is debouncing

If the current event is timemout for debounce

Change state to ready to sample  
 Else if the button state is ready to sample

If the current event is button down

Re-initialize the 200ms timer

Change the button state to button debouncing

}

# Psuedo code for control sensors

bool InitializeControlSensors ( uint8\_t Priority )

{

Initialize the Analog input ports

Init cycle timer

}

ES\_Event RunControlSensors( ES\_Event ThisEvent )

{

If the cycle timer is time out

Read the analog and put them into an array

}

}

Query the vertical Analog value function

Query the horizontal Analog value function

Query the potentiometer Analog value function

Query the coin sensor Analog value function