

Overview

Initialization functions set the appropriate registers and also create the servo library.

Event Processor processes EventCheckers events , which are primarily for the buttons/roamer select and timers

sendSM and receives are small state machines that take care of everything related to sending and receiving messages. They use either a count or a state to keep track of what byte to send next and also include all the error checking.

Master Controller

Global variables:

char array – transmit (keeps information for sending)

char array – receive(keeps received information)

int – currentRoamer (keeps track of current connected roamer number)

int – DestinationMSB, DestinationLSB (keeps track of current connected roamer XBee)

boolean – receiving, sending, connecting, disconnecting, broadcast, connected (to save state of machine)

InitComm

Set RE, RIE, TE flags in SCI1CR2

Set SCI1BDH/L to 156

InitTimers

Set timers 0 and 1 to 200 ms and start them

Set enable on TSCR1, and rate multiplier to 8 for 3MHz

Set rise on compare for all channels (OLx and OMx high)

Set toggle on overflow (TOVx)

Set TCx to 0xFFFF – 4500 to start, which gives a servo neutral position of 1.5 ms pulse width

RunMasterSM (Event processor)

If timer 0 (for 5 Hz sending) timed out,

set sending to true

transmit interrupt enable to high

if connecting, disconnecting, or already connected, set timer 0 to 200 and start it

If timer 1 timed out (for debouncing buttons)

Post event to ButtonSM

If EventType is RoamerSwitch

Set currentRoamer to Roamer number

If connected

 If currentRoamer is not the same as last roamer

 Set connecting to true, and set timer 0 to 200 ms and start it

If not connected

 Set connecting to true and set timer 0 to 200ms and start it

If connected, set disconnecting to true

If EventType is ButtonDown

 Toggle camera (true/false)

sendByte (returns a boolean, takes a char msg input)

if TDRE flag is high, send message by writing it to SCI1DRL and return true

else return false

sendSM

States: TX_START, TX_MSBLength, TX_LSBLength, TX_API, TX_FRAME, TX_MSBDestination, TX_LSBDestination, TX_OPTIONS, TX_FIRST, TX_LEFT, TX_RIGHT, TX_GRIPPER, TX_CAMERA, TX_DIGITAL, TX_CHECKSUM

For details, look at communications protocol

For each state –

Set byteToSend to an 8 bit char (Varies based on state)

If sendByte(byteToSend) returns true,

Set CurrentTXState to next state in line, listed above

Add byteToSend to checksum

sendByte for each state –

TX_Start – 0x7E

TX_MSBLength – 0x00

TX_LSBLength – 0x0B

TX_API – 0x01

TX_FRAME – 0x01, incrementing each time, if frameID goes above 50, set back to 1

TX_MSBDestination – if connecting or broadcast, set to 0xFF

Else set byte to DestinationMSB

TX_LSBDestination – if connecting or broadcast, set to 0xFF

Else set byte to DestinationLSB

TX_OPTIONS – 0x00

TX_FIRST – if connecting, 0x00, if disconnecting, 0x01, else 0x02

TX_LEFT – read accelerometer inputs, normalize by gravity voltage, multiply by scaling factors for angle (.7 for square root of 2 divided by 2 so 45 degrees will put it on full drive), and create ints left and right based on gravity measurements.

If disconnecting, 0x00, if connecting, currentRoamer, else send left

TX_RIGHT - If disconnecting or connecting, 0x00, else send right

TX_GRIPPER – if disconnecting or connecting, 0x00, else

Read ADPins for IR distance sensors. If both are above 500 (~2.7V), send 0x80 for closed.

If both are below 500, send 0x00 for empty. If neither are true, send the lastGripperState

TX_CAMERA – if disconnecting or connecting, 0x00, else if camera is true, send 0x80, else send 0x00.

TX_DIGITAL - if disconnecting or connecting, 0x00, else send PTU bits 4, 5, and 6 in bits 1, 2, and 3 (PTU & 0x70) >> 4)

TX_CHECKSUM – (0xFF – (checksum & 0xFF))

After sending, if connecting or disconnecting or connected, increment error count

If error count > 5, set connected, connecting, disconnecting to false, DestinationMSB, DestinationLSB, error count to 0.

receiveSM

For details, look at communications protocol

Static int I (count) keeps track of state

For all states - Set input to SCI1DRL, and set receive[i] to input

Important distinctions-

Static Boolean active keeps track of whether or not we are in a message.

If input is a HeaderByte (0x7E), and active is low,

Set active to high, i to 2, receiving to true

After receiving length, check if length < 1 or > 100. If it is, set it to 3. (Length protection)

For APIIdentifiers of 0x81 – Set SourceMSB, LSB based on bytes 5 and 6. If connected is true and the sourceLSB and MSB do not match DestinationLSB and MSB, set active to 0.

On Status byte, if disconnecting is true and received byte is 0x05 and Source matches Destination, set disconnecting to false, connected to false, Destination to 0.

If connecting is true and received byte is 0x04, set connected to true and light LED corresponding with roamer number. Set connecting to false and Destination to Source.

On Battery byte, set Servo to Battery*11/17 (scales 255 to 180)

On Camera byte, save input to lastCamera.

Interrupt response

If RDRF flag in SCI1SR1 is high, call receiveSM

Else if TDRE flag is high and sending is true, call sendSM

EventCheckers

For each button,

Save current state to a variable.

If QueryButtonSM returns READY2SAMPLE,

 Post START_DEBOUNCE to ButtonSM

 If CurrentState of button is not equal to last state of button,

 Send corresponding event to Master (ButtonDown, RoamerSwitch with Param = 1,2,3)

Set last state of button to current state of button

ButtonSM

If timer 1 times out, set current state to READY2SAMPLE

else if event is START_DEBOUNCE and Param is 1,

 set timer 1 to 50 ms and start it

 set CurrentState to DEBOUNCING

QueryButtonSM

Return ButtonSM CurrentState