

## main.c – PIC main entry point pseudo code

---

### function declarations

```
static void init(void);
static char balloonPopped(void);
void main(void);
```

### referenced global variables

```
extern unsigned int g_overflows;           // timer0 overflows
extern unsigned int g_overflows2;        // timer2 overflows
extern unsigned int g_duty[];            // duty cycle for each servo channel
extern char g_servoPortState[];         // PORTC line state
```

### module variables

```
static unsigned int r2_overflows = 0; // tells when to turn r2d2
```

### interrupt service routine

```
void interrupt mainHSV_ISR(void); // isr for timer0, timer2, ccp1 (output compare)
```

---

### pseudocode

---

---

```
void main (void)
    initialize things                               // init()

    while (1)
        // check whether or not to transmit anything
        if (readyToTransmit()) {
            // get byte to send
            send_byte = getTransmitByte();
            // send it
            transmit(send_byte);
        }
        // end transmit
```

```

// check whether or not ready to receive
if (readyToReceive()) {
    // get receive byte
    receive_byte = receive();
    // put the receive byte somewhere
    placeReceiveByte(receive_byte);

    // if full packet received, actuate things
    if (fullPacketReceived()) {
        // actuate from resulting packet

        // 1. set throttle and direction
        // if human ...
        if (getIsHuman()) {
            set throttle (no performance reduction)
            set direction (no performance reduction)
        }
        else {
            set throttle (with performance reduction)
            set direction (with performance reduction)
        }

        // 2. shoot or dont
        set shoot pin = getShoot()
        // if commanded
        if (getShoot()) {
            start shoot timer
            deploy shoot servo
        }

        // 3. display paired status
        set isPaired LED indicator
        if (getIsPaired()) {
            don't spin r2d2
        }

        // 4. display zombie status
        set / clear human/zombie status led indicator

        // by here we've processed the packet
        let state machine know we've processed the packet
    }
}
// end receive

// if connection timer expires, clear buffers and reset
if ( isTimerExpired() ) {

```

```

        carry out connection time expired routine
        set isHuman indicator led
        clear isPaired indicator LED
        Zero throttle and direction duty cycles
        make r2 spin around
        undo shoot
        reset shoot timer

    }

    // if shoot timer is expired
    if (isTimerExpired2() ) {
        undo shoot servo
        reset shoot timer
    }

    // if balloon pops and we're human
    if ( balloonPopped() && getIsHuman() ) {
        notify HSV state machine that were not human anymore
        reset HSV state machine
        Zero throttle and direction
        turn off isHuman LED indicator
        turn off isPaired LED indicator
        make r2 spin around
        undo shoot
        reset shoot timer
    }

    // r2 spin here
    if (r2SpinFlag) {
        spin R2D2 back and forth
    }

    end while

end main
-----

void interrupt mainHSV_ISR(void)

    if (output compare interrupt flag is set)
        first3 = status of pins that aren't servo channels
        PORTC = first3 + nextServoPinState

        record timer1 low value

```

```
record timer1 high value

// find next value to put in ccp1 registers
if (thisState != wait state)
    thisDuty = g_duty[thisState]
end if
else
    thisDuty = (sum of all g_duty)
end else

ccp1rl = low timer value + low byte of thisDuty
ccp1rh = high timer value + high byte of thisDuty

increment thisState (but don't let it get larger than 5)

clear output compare interrupt flag

end if

if (timer 0 overflow flag is set)
    increment g_overflows
    clear timer0 overflow interrupt flag
end if

if (timer 2 overflow flag is set)
    increment g_overflows2
    clear timer2 overflow interrupt flag
end if
end mainHSV_ISR
```

---