```c
/*
 * L2P8.c
 *
 * Created: 2/29/2020 4:01:11 AM
 * Author : Victor Lam
 */

#include <avr/io.h>

void SPI_MasterTransmit(uint8_t data) {
    SPDR = data; // Start transmission
    while(!(SPSR & (1<<SPIF))); // Wait for transmission complete
    PORTB ^= (1<<PB2);
    PORTB ^= (1<<PB2);
}

int getDigit(uint8_t last) {
    uint8_t next = 0;
    if (last == 2) { //2
        next = 0b11011010;
        } else if (last == 3) { //3
        next = 0b11110010;
        } else if (last == 4) { //4
        next = 0b01100110;
        } else if (last == 5) { //5
        next = 0b10110110;
        } else if (last == 6) { //6
        next = 0b10111110;
        } else if (last == 7) { //7
        next = 0b11100000;
        } else if (last == 8) { //8
        next = 0b11111110;
        } else if (last == 9) { //9
        next = 0b11100110;
        } else if (last == 1) { //1
        next = 0b01100000;
        } else { //then 0
        next = 0b11111100;
    }
    return next;
}

int getDD(uint8_t last) {
    uint8_t next = 0;
    if (last == 2) { //2
        next = 0b11011011;
        } else if (last == 3) { //3
        next = 0b11110011;
        } else if (last == 4) { //4
```

```
        next = 0b01100111;
        } else if (last == 5) { //5
        next = 0b10110111;
        } else if (last == 6) { //6
        next = 0b10111111;
        } else if (last == 7) { //7
        next = 0b11100001;
        } else if (last == 8) { //8
        next = 0b11111111;
        } else if (last == 9) { //9
        next = 0b11100111;
        } else if (last == 1) { //1
        next = 0b01100001;
        } else { //then 0
        next = 0b11111101;
    }
    return next;
}

void setDisplay(int j, int t) { //sets a display to output and transmits a byte
    if (j == 0) { //if count == 0
        PORTC = (1<<PC2); //set PC2 to high: display all the way on the left
        j = (t / 1000) % 10; //Isolate digit all the way on the left
        SPI_MasterTransmit(getDD(j)); //transmit the display byte associated with this
        } else if (j == 1) {
        PORTC = (1<<PC3);
        j = (t / 100) % 10;
        SPI_MasterTransmit(getDigit(j));
        } else if (j == 2) {
        PORTC = (1<<PC4);
        j = (t / 10) % 10;
        SPI_MasterTransmit(getDD(j));
        } else if (j == 3) {
        PORTC = (1<<PC5);
        j = t % 10;
        SPI_MasterTransmit(getDigit(j));
    }
}

void startStop() {
}

int main(void){
    DDRD |= ~(1<<PD0) | (1<<PD3);
    PORTD |= (1<<PD0) | (1<<PD3);
    DDRB = (1<<PB3)|(1<<PB5)|(1<<PB2); // Set MOSI and SCK output and ss, all others
      input
    SPCR = (1<<SPE)|(1<<MSTR)|(1<<SPI2X); // Enable SPI, Master, set clock freq to 1
      MHz
```

```
    TCCR1B |= (1<<CS11);
    DDRC = (1<<PC5)|(1<<PC4)|(1<<PC3)|(1<<PC2);
    uint16_t target = 0x00FA;
    int count = 0;
    int time = 0;
    int tick = 0;
    int ss = 0;

    label:
    while (1) {
    //  label:
        if (TCNT1 == target) { //1ms
            target += 0x00FA;
            count++;
            tick++;
            if (tick == 100) {
                tick = 0;
                time++;
            }
            if ((time / 100) % 10 == 6) time += 400;
            SPI_MasterTransmit(0b00000000);
            setDisplay(count, time);
            if (count == 3) count = -1;
            if ((PIND & (1<<PD0)) == 0) time = 0;
            if ((PIND & (1<<PD3)) == 0) goto ladel;

        } //End of if
    } //End of while (1)
    ladel:
    while (1) {
        if (TCNT1 == target) { //1ms
            target += 0x00FA;
            count++;
            SPI_MasterTransmit(0b00000000);
            setDisplay(count, time);
            if (count == 3) count = -1;
            if ((PIND & (1<<PD0)) == 0) time = 0;
            if ((PIND & (1<<PD3)) == 0) goto label;
        }
    }
}
```