//Matthew Currey

//Embedded Systems

//FINAL PROJECT

#include <msp430.h>

int count;

int callforhelp;

int alarm;

alarm = 0; //0 is unarmed, 1 is armed/blinking light, 2 is solid light/ calling 911

callforhelp = 0; //Counter when smoke is detected. calls 911 when = 20

count = 0; //\*IN FUTURE VERSION: ADD REST TIMER AND USE COUNT AS COUNTER UNTIL SENSOR WORKS AGAIN\*

void TimerA3Setup();

void Setupinit();

void main()

{

// WDTCTL = WDTPW | WDTHOLD; // stop watchdog timer

WDTCTL = WDT\_ADLY\_1000; //ASK SOMEONE!!!!!!!

Setupinit();

PM5CTL0 &= ~LOCKLPM5;

\_\_bis\_SR\_register(LPM0\_bits | GIE); // Enter LPM0 w/ interrupt waiting for interrupt

\_\_no\_operation(); // For debugger

}

void Setupinit()

{

P1DIR |= 0x40; //sensor

P1OUT |= 0x08; // sensor

P1REN |= 0x08; //sensor

P1IES &= ~BIT4; // P4.1 Low to High

P1IE |= BIT4; // P4.1 interrupt enabled

P1IFG&= ~BIT4;

P3DIR |= BIT0; //3.0 MAKES LED BLINK & BUZZER TO SOUND

P3OUT &= ~BIT0; //CLEAR

P2OUT |= BIT0; // Configure P2.1 as pulled-up

P2REN |= BIT0; // P2.0 pull-up register enable

P2IES &= ~BIT0; // P2.0 Low to High

P2IE |= BIT0; // P2.0 interrupt enabled

P4DIR |= BIT0; //4.0 SENDS a signal to WIFI MODULE when callforhelp = 20

P4OUT &= ~BIT0;

}

#pragma vector=PORT1\_VECTOR

\_\_interrupt void Port\_1(void)

{

//WHEN THE SENSOR GOES OFF TIMER 3 BEGINS HERE

TB3CCTL0 |= CCIE; // Enable TB0 CCR0 Overflow IRQ

TB3CCR0 = 0xFFFF; // PWM Period

TB3CCTL1 = OUTMOD\_7; // CCR1 reset/set

TB3CCR1 = 800; // CCR1 PWM duty cycle

TB3CTL = TBSSEL\_\_SMCLK | ID\_3 | MC\_\_UP | TBCLR; // SMCLK, up mode, clear TBR

alarm = 1; //ARMED

P1IFG&= ~BIT4;

}

#pragma vector=TIMER3\_B0\_VECTOR

\_\_interrupt void TIMER3\_B0\_ISR(void)

{

//TIMER 3

if (alarm == 1)

//IF SENSOR JUST WENT OFF AND BUTTON HASN"T BEEN PRESSED

if (callforhelp == 20)

// IF callforhelp = 20 (LED blink 20 times)

{

P3OUT |= BIT0; //LIGHT STAYS ON

P4OUT |= BIT0; //SEND SIGNAL TO WIFI MODULE THAT TEXTS PHONE

alarm = 2; //alarm = 2: solid light/calling 911

}

else

//IF callforhelp <> 20 (LED hasn't blinked 20 times)

{

P3OUT ^= BIT0; //BLINK LIGHT & SOUND BUZZER

callforhelp++; //ADD COUNTER for how many times led blinked (when callforhelp = 20: text phone)

}

}

}

#pragma vector=PORT2\_VECTOR

\_\_interrupt void Port\_2(void)

//Interrupt for the button

{

///buttons have debouncing but we dont have to take that into consideration because all the button does is turn everything off

P3OUT &= ~BIT0; //Clear light and buzzer

P4OUT &= ~BIT0; //Clear any signal to WIFI MODULE

//RESETS TIMER, SETTING EVERYTHING to 0, TURNING IT OFF

TB3CCR0 = 0x00; // PWM Period

TB3CCTL1 = OUTMOD\_7; // CCR1 reset/set

TB3CCR1 = 0;

// rest all counters

callforhelp = 0;

alarm = 0; //Unarmed

P2IFG&= ~BIT0;

}