Interrupts, Timers, and more

Michael Caruana

RBT211 – Arduino Embedded Programming

George Muhn – Professor

Engineering Notebook 4

The code below is what I used in my ‘Timers’ solution in Microchip. It took a little bit to get it correct but I finally figured it out while watching the class again. This assignment was rather difficult for me. I needed to watch the class that I was a part of again to catch the instruction. Sometimes I have to watch the videos several times to grasp the concepts but that is how I learn.

I have a screenshot of the code in Microchip, and I also have pictures of the program working on the breadboard. There is also a link to a short 2-minute video showing that the program works.

Obviously, I had copied and pasted code but to get it to work correctly I had to make sure it was organized correctly. I think the first time I built it; I had some 8 or 10 errors that I quickly corrected. I was very happy when the project built successfully

<https://youtu.be/SH-7WBUb2HI> - video

/\*

\* Timer.c

\*

\* Created: 22-Jun-23 19:58:02

\* Author :Michael Caruana

\*/

//Always include the F\_CPU to reduce errors

#ifndef *F\_CPU* // if F\_CPU was not defined in Project -> Properties

#define *F\_CPU* 16000000UL // define it now as 16 MHz unsigned long

#endif

#include <avr/io.h> // this is always included in AVR programs

#include <avr/interrupt.h> //Required Libraries for interupt

#define LED PB2

int main(void)

{

DDRD &= ~(1 << PD2); //Clear pin PD2 for input

PORTD |= (1 << PD2); //Set the internal pull-up resistor

DDRD |= (1 << PD4); //Set pin PD4 as output

DDRB |= (1<< LED);

TCCR1A = 0b00000000;

TCCR1B = 0b00001101;

TCCR1C = 0b00000000;

TIMSK1 = 0b00000010;

OCR1A = 15625;

//External Interrupt Set up

// Check which External interupt pin you want to use.

EICRA |= (1<< INT0);

EIMSK |= (1<< ISC00);

sei();

/\* Replace with your application code \*/

while (1)

{

//No code here

}

return(0);

}

ISR(TIMER1\_COMPA\_vect){

PORTB ^= (1<< LED);

}

ISR(INT0\_vect){

if(!(PIND & (1 << PD2))){

PORTD |= (1<<PD4);

}else if (PIND & (1 << PD2)){

PORTD &= ~(1<<PD4);

}else { //do nothing

}

}

/\* You may define your pins ports and DDR here

// EXAMPLE

// #define LED0 PB0 // This will set variable LED0 as PB0

int main(void)

{

//SET UP PINS INPUT, OUTPUTS AND PULL UP/DOWN RESISTORS HERE:

//

//Timer Set up HINT: NOT ALL TIMERS WILL BE USED AND MAY BE REMOVED, COMMENTED LEFT as 0

//Provided Timer 0 8bit timer Registers

TCCR0A = ;

TCCR0B = ;

TIMSK0 = ;

//Provided Timer 1 16bit timer registers

TCCR1A = ;

TCCR1B = ;

TCCR1C = ;

TIMSK1 = ;

//Provided Timer 3 8bit timer register

TCCR2A = ;

TCCR2B = ;

TIMSK2 = ;

\*/

// Timer Interrupt Code, What to do once the timer has reached its count or overflow

// Tassignment will require 2 interupt ISR() functions

// ISR(/\*INSERT TYPE OF INTERUPT\*/) {

// INPUT CODE TO BLINK LED HINT: Look at the blink led assignment

//}

/\*TYPES OF INTERUPTS

1. INT0\_vect

Looks for interrupt flag on external interrupt pin 0

2. INT1\_vect

Looks for interrupt flag on external interrupt pin 1

3. TIMER(0,1 or 2)\_COMPA\_vect

Looks for timer comparison A count on selected timer 0,1, or 2

4. TIMER(0,1 or 2)\_COMPB\_vect

Looks for timer comparison B count on selected timer 0,1, or 2

5. TIMER(0,1 or 2)\_OVF\_vect

Looks for Overflow of timers 0 1 or 2

\*/

A picture containing text, screenshot, display, computer

Description automatically generated

A picture containing text, screenshot, software, display

Description automatically generated

Pressing the button to make the Blue LED turn on.
 Pressing the button to make the Blue LED turn on while the Yellow LED flashes for one(1) second.


Description automatically generated

Finger on the button, ready to push.

Description automatically generated