Háskólinn í Reykjavík

Embedded System Programming

Assignment 1

Hand-in

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

By:
Andri Þór Arnarsson
andrita20@ru.is
Zeyad Rafat Surakji
ziad19@ru.is

Instructor:
Torfi Þórhallsson
torfith@ru.is

August 28, 2020

"The nitrogen in our DNA, the calcium in our teeth, the iron in our blood, the carbon in our apple pies were made in the interiors of collapsing stars. We are made of starstuff."

- Carl Sagan

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

1 Introduction

Part 1

Load and try out the diode blinking sample program.
Replace the delay() function with your own busyWait() function accepting the same argument.
Inside this function you can use a double for-loop, where the inner loop contains a single integer (int)
addition, e.g. c += 9;
Adjust the loop counter to get similar behaviour to delay(). You can disable optimization as in Assignment 1.1.
What effect does it have to use multiplication or division instead of addition? What about using a 64 bit integer or a double?

Part 2
Control the blink rate using an analog input. You can use any sensor or circuit that changes the value of the analog input.
Make a short video showing the sensor controlled blinking in action. Upload the video to YouTube and submit a link.
We aim to upload the code to a git repository. This we will cover next week.
This is a group assignment.

2 Part 1

2.1 Footprint Results

2.1.1 One for loop normal int with $*=$

- RAM: [ ] 0.0% (used 0 bytes from 2048 bytes)
- Flash: [ ] 0.8% (used 256 bytes from 30720 bytes)

2.2 double as parameter in for loop

- RAM: [ ] 0.0% (used 0 bytes from 2048 bytes)
- Flash: [ ] 3.2% (used 998 bytes from 30720 bytes)

2.3 64b uint as parameter in for loop

- RAM: [ ] 0.0% (used 0 bytes from 2048 bytes)
- Flash: [ ] 2.0% (used 614 bytes from 30720 bytes)

2.4 Code Part 1

2.4.1 main.cpp

```
/*
 * Assignment Project 1
 * Authors : Andri Þór Arnarsson
 *         : Zeyad Rafat Surakji
 *         : ...
*/
#define F_CPU 16000000UL
#include <stdio.h>
#include <avr/io.h>
#include <util/delay.h>
#include "digital_out.h"


int main(void) {

   DigitalOut led(5);
   led.init();
        int input = 0;

        while(1) {
           led.toggle();
       busyWait(1000);

        }

        return 0;
}
```

2.4.2 digital_out.h

```
/*
 * Project 1
 * Authors : Andri Þór Arnarsson
 *         : Zeyad Rafat Surakji
 *         : ...
*/

#ifndef DIGITAL_OUT_H
#define DIGITAL_OUT_H

#include <stdint.h>

// Class Prototypes
class DigitalOut
{
public:
    DigitalOut(int pin);
    void init();
    void setHi();
    void setLo();
    void toggle();
private:
    uint8_t pinMask;
};

// Function Prototypes
void busyWait(int wait);


#endif // DIGITAL_OUT_H
```

2.4.3 digital_out.cpp

```
/*
 * Assignment 1.1 in C
 * Authors : Andri Þór Arnarsson
 *         : Zeyad Rafat Surakji
 *         : ...
*/
#include "digital_out.h"
#include <avr/io.h>
#include <stdint.h>

DigitalOut::DigitalOut(int pin){ pinMask = (1 << pin); }

void DigitalOut::init(){ DDRB |= pinMask; }

void DigitalOut::setHi(){ PORTB |= pinMask; }

void DigitalOut::setLo() { PORTB &= ~pinMask; }

void DigitalOut::toggle(){ PORTB = PORTB ^ pinMask; }

void busyWait(int wait){
    wait *= 16000; // 1/16mhz * 16000 = 0.001s not precise but hopefully close enough
    for(int i = wait ; i <= wait ; i++ ){
        i++;
    }
}
```

3 Part 2

Idea was to use a Potentiometer to control the the voltage on the ADC and the blinking of the LED

3.1 Footprint Results

- RAM: [ ] 0.1% (used 2 bytes from 2048 bytes)
- Flash: [ ] 2.5% (used 768 bytes from 30720 bytes)

3.2 Code Part 1

3.2.1 main.cpp

```
/*
 * Project 1
 * Authors : Andri Þór Arnarsson
 *         : Zeyad Rafat Surakji
 *         : ...
*/
 #define F_CPU 16000000UL
 #include <stdio.h>
 #include <avr/io.h>
 #include <avr/interrupt.h>
 #include "digital_out.h"
 #include "adc_init.h"

 int main(void) {

        adc_setup();
    adcConvert();
    sei();
    DigitalOut led(5);
    led.init();

         while(1) {
        busyWait(adc_result);
            led.toggle();
         }

         return 0;
 }

ISR(ADC_vect)
{
    adc_result = ADC;
    adcConvert();

}
```

3.2.2 adc_init.h

```h
/*
 * Project 1
 * Authors : Andri Þór Arnarsson
 *         : Zeyad Rafat Surakji
 *         : ...
*/
#ifndef ADC_INIT_H
#define ADC_INIT_H
#include <stdint.h>

uint16_t adc_result;
void adc_setup(void);
void adcConvert();
#endif // ADC_INIT_H
```

3.2.3 adc_init.cpp

```cpp
/*
 * Project 1
 * Authors : Andri Þór Arnarsson
 *         : Zeyad Rafat Surakji
 *         : ...
*/

#include <avr/io.h>


void adc_setup(void){

    // Voltage Reference Selections for ADC
        // Select Vref = AVcc
    // ADC pin5 PC45
    ADMUX  = ( 1<<REFS0 | 1 << MUX0 | 1 << MUX2);

    // ADC Control and Status Register A
    // ADEN - ADC Enable
    // ADIE - ADC Interupt Enable
    // ADPSx - Prescaler = 128
        ADCSRA = (1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0)|(1<<ADIE)|(1<<ADEN);
    // Disable digital input buffer
    DIDR0 = ( 1 << ADC5D);
}

void adcConvert(){
    ADCSRA |= ( 1 << ADSC );
}
```