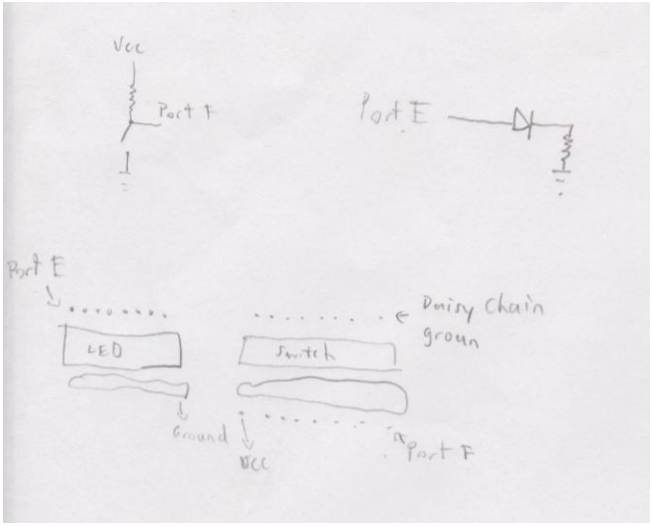
Lab 2

Reyna, Ricardo

Section 75C9

06/03/2015

- Prelab Questions:
  None
- Problems Encountered:
  In KITT2 there's a double delay at 0x18.
- Future Work/Applications:
  This lab will help me interact with the input and output ports
- Schematics:



- Pseudo code:
  Part a:
  Check PORT F
  load value into register
  load register into PORT E

  Part b:
  Have an ON and OFF subroutine
  LOOP with a bunch of nop and a counter
  Bounce between both subroutines

  Part c:
  KITT1:
  set carry
  rotate
  delay
  if 0xFF restart
  check for bit 3

KITT2:
load 0x81
load a register with 0x80
other register with 0x01
shift right first register
shift left other register
store the first one in a temp
or both register
delay
load temp into first regster
check for kitt1

- Program Code:
  PartA:

```
/*
 * Lab2_A_RSR.asm
 *
 *  Created: 5/28/2015 4:06:55 PM
 *   Author: stefano92
 Lab 2 Part a
 Name: Ricardo Stefano Reyna
 Section#: 75C9
 TA: Khaled Hassan
 Description: This program writes to port E and reads from port F
 */

.NOLIST
.include "ATxmega128A1Udef.inc"

.LIST
.org 0x0000          ;code starts at this address
       rjmp MAIN     ;go to MAIN

.org 0x200
MAIN:
       sts PORTF_DIR, r16          ;set port F as input
       lds r16, PORTF_IN           ;load value of port F into r16
       sts PORTE_DIRSET, r16       ;Set port E as output
       sts PORTE_OUT, r16          ;send value of r16 to port E
       rjmp MAIN                          ;go to main
```

PartB:

```
/*
 * Lab2_B_RSR.asm
 *
 *  Created: 5/31/2015 11:33:45 PM
 *   Author: stefano92
 Lab 2 Part b
 Name: Ricardo Stefano Reyna
 Section#: 75C9
 TA: Khaled Hassan
 Description: This program creates a delay for 200us
```

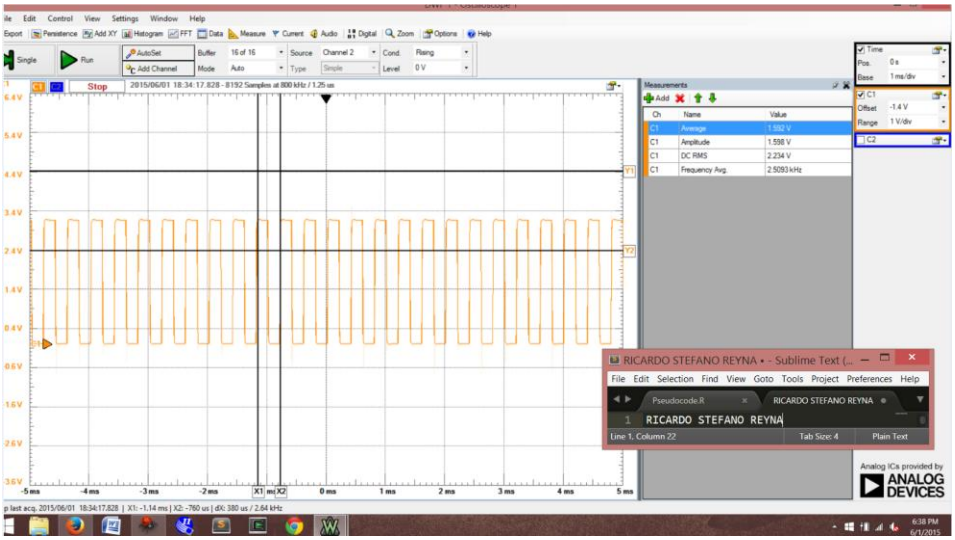
```
 */

 .NOLIST
.include "ATxmega128A1Udef.inc"

.LIST
.org 0x0000          ;code starts at this address
       rjmp MAIN     ;go to MAIN

.org 0x200
MAIN:
       ldi r16, 0x00 ;load 0 into r16
       ldi r17, 0x01 ;load 1 into r17
       ldi r19, 0x00 ;load 0 into r19
ON:
       sts PORTE_DIRSET, r17       ;Set Port E as output
       sts PORTE_OUT, r17          ;give port E the value of r17
       ldi r18, 0x00               ;reset r18
       inc r19                                  ;increase r19
LOOP:
       nop                  ; do nothing
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       inc r18              ;increase r18
       cpi r18, 28   ;check if r18 is 28
       brne LOOP     ;if not loop
       cpi r19, 0    ;check if r19 is 0
       breq ON              ;if true then go to ON
OFF:
       sts PORTE_DIRSET, r16       ;Set port E as output
       sts PORTE_OUT, r16          ;port E contains the value of r16
       ldi r18, 0x00               ;reset r18
       dec r19                                  ;decrease r19
       rjmp LOOP                        ;go to LOOP

       PartC:

/*
 * Lab2_C_RSR.asm
 *
 *  Created: 6/1/2015 6:53:38 PM
 *   Author: stefano92
 Lab 2 Part c
 Name: Ricardo Stefano Reyna
 Section#: 75C9
 TA: Khaled Hassan
 Description: This program will have two settings where depending on Port F's bit 3 it
will run
 KITT1 or KITT2
 */
```

```
.NOLIST
.include "ATxmega128A1Udef.inc"

.LIST
.org 0x0000          ;code starts at this address
       rjmp MAIN     ;go to MAIN

.org 0x200
MAIN:
       ldi r22, 0x00               ;load r22 with 0
       ldi r21, 0x00               ;load r22 with 0
       sts PORTF_DIR, r20          ;set port F as input
       lds r20, PORTF_IN           ;load value of port F into r20
       andi r20, 0x08                     ;and input with 0x08
       cpi r20, 0x08               ;check for bit 3
       brne KITT1                         ;if false go to KITT1
       jmp KITT2                          ;else go to KITT2
KITT1:
       ldi r16, 0xFE
       sts PORTE_DIRSET, r16       ;Set port E as output
       sts PORTE_OUT, r16          ;port E contains the value of r16
       sec                                      ;this is to do a rotate with no carry
LOOP1:
       ROR    r16
EXTRA1:
       inc r21
       cpi r21, 74
       brne DELAY_10ms1
       rjmp CONT1
DELAY_10ms1:
       nop                  ; do nothing
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
       nop
```

```
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        inc r22              ;increase r18
        cpi r22, 0xFF ;check if r18 is 28
        brne DELAY_10ms1     ;if not loop
        rjmp EXTRA1
CONT1:
        ldi r22, 0x00
        ldi r21, 0x00
        cpi r16, 0xFF
        brne NEXT1
        ldi r16, 0x7F
        sec
NEXT1:
        sts PORTE_DIRSET, r16       ;Set port E as output
        sts PORTE_OUT, r16          ;port E contains the value of r16

        lds r20, PORTF_IN           ;load value of port F into r16
        andi r20, 0x08
        cpi r20, 0x08
        breq KITT2
        rjmp LOOP1


/*
;
;BREAK POINT OF THE PROGRAMS
;
*/

KITT2:
        ldi r16, 0x81
        sts PORTE_DIRSET, r16       ;Set port E as output
        sts PORTE_OUT, r16          ;port E contains the value of r16
        ldi r16, 0x80
        ldi r17, 0x01
        ldi r19, 0x00               ;used for checking
LOOP2:
        lsr r16                                 ;shift right
        lsl r17                                 ;shift left
        mov r18, r16                ;temp
        or r16, r17
EXTRA2:                                         ;This is used to multiply the delay_10ms
        inc r21
        cpi r21, 146
        brne DELAY_10ms2
        rjmp CONT2
DELAY_10ms2:
        nop                  ; do nothing
        nop
        nop
        nop
        nop
```

```
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        inc r22                             ;increase r18
        cpi r22, 0xFF           ;check if r18 is 28
        brne DELAY_10ms2        ;if not loop
        rjmp EXTRA2
CONT2:
        sts PORTE_DIRSET, r16       ;Set port E as output
        sts PORTE_OUT, r16          ;port E contains the value of r16
        cpi r16, 0x81
        brne NEXT2
        cpi r19, 0x00
        brne KITT2
NEXT2:
        inc r19
        mov r16, r18
        ldi r22, 0x00
        ldi r21, 0x00
        lds r20, PORTF_IN           ;load value of port F into r16
        andi r20, 0x08
        cpi r20, 0x00
        brne LOOP2
        jmp KITT1
```

- Appendix:
  PartB:


Stop
Measurements
Ch Name Value
C1 Average 1.592 V
C1 Amplitude 1.598 V
C1 DC RMS 2.234 V
C1 Frequency Avg. 2.5093 kHz
RICARDO STEFANO REYNA • - Sublime Text (...
File Edit Selection Find View Goto Tools Project Preferences Help
Pseudocode.R
RICARDO STEFANO REYNA
1 RICARDO STEFANO REYNA
Line 1, Column 22
Tab Size: 4
Plain Text
Analog ICs provided by
ANALOG DEVICES