ECE 367 - Experiment #1

Spring 2006 Semester

Introduction

The goal of this experiment is the acquaint you with the Technological Arts MicroStamp11 68HC11 microcontroller development system, and with the procedure used to assemble/download/execute code on this processor. Specifically, you will do what is needed to have the microcontroller independently execute code that blinks an LED. To do this you will need to intall two programs on a Windows PC and have access to its serial port.

Required Hardware and Software

- MicroStamp11 starter kit (UIAC1) from Technological Arts
- 9v battery
- THRSim11 68HC11 Simulator (CD-ROM from the Spasov textbook)

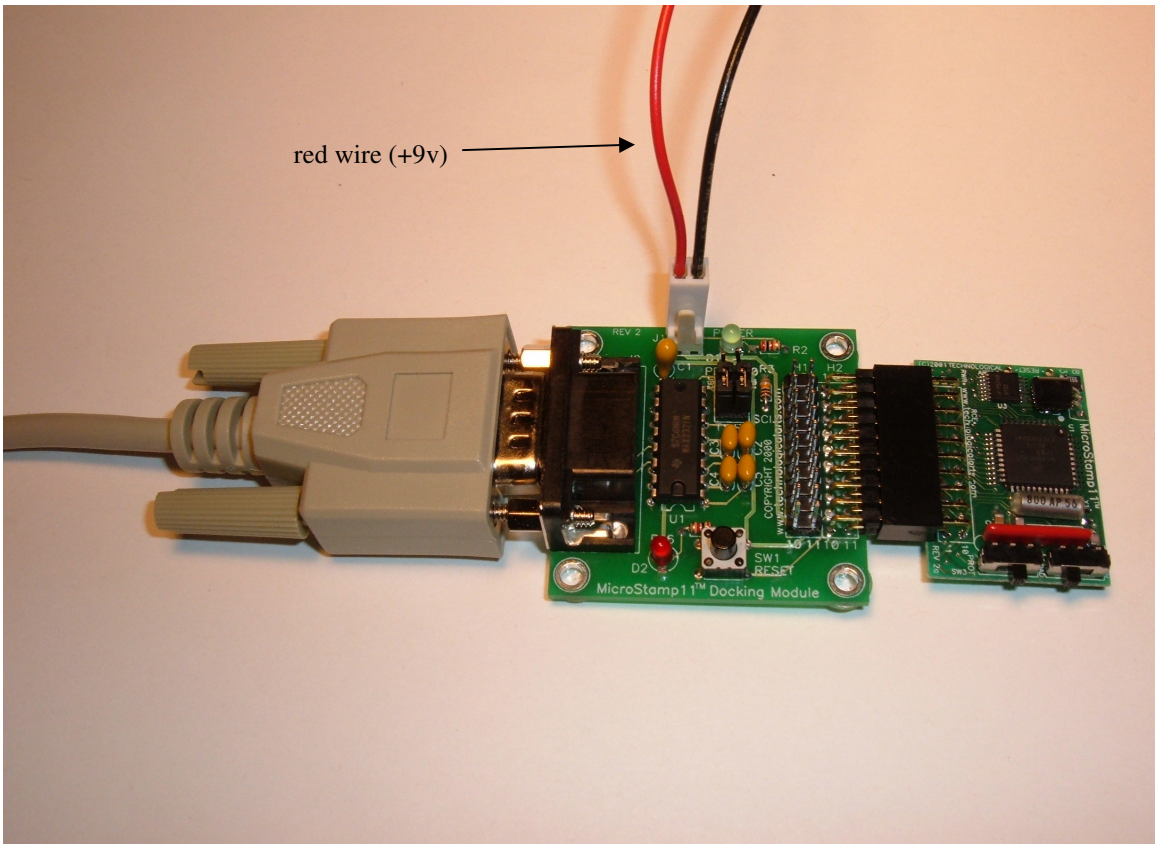
Procedure

1. Install the THRSim11 assembler/simulator program from the CD-ROM that came with your textbook.

2. Using THRSim11, assemble the flashing-LED code that is provided for you in the Appendix (you may copy and paste from an on-line listing of this program that is found at http://www.ece.uic.edu/~goncharo/ece367_exp1.txt). A simulation window will open having the first line of executable code highlighted in green - close this window, we will learn to simulate code later. By assembling the program, an executable code having *.s19 extension should have been created in the same directory (e.g. "C:\Program Files\THRSim11\sdemo\ece367_exp1.S19").

3. Install the MicroLoad program from the CD-ROM that came with your MicroStamp11 starter package.

4. Connect your MicroStamp11 module to its docking station. The component sides of the two boards should both face upward, as shown in the photo. Next you need to power up the board. The easiest method is to connect the terminals of a 9 Volt battery to the power cable that is provided (observe proper polarity: red +, black −) and plug it into the docking module. If done correctly a green power indicator LED will light brightly. CAUTION: if either the battery or the power cable is hooked up backwards, you will probably ruin the microcontroller (whose replacement cost is $34 plus shipping). Always check and doublecheck your power supply connections! Refer to the photo below.



Run MicroLoad and in the options menu configure it to work with MicroStamp11, standard version. Also specify the COM port that your PC gives you access to. Connect the other end of the serial cable to this serial communications port. Next complete the downloading of code by specifying your file name (in the THRSim11 directory, having .s19 extension) and follow directions regarding the two mode switch settings and reset switch when downloading the code.

If it appears that code is being transferred but then stops before fully downloaded, try the following: lay everything down on a desk and don't touch it while code is downloading. This prevents glitches from occurring due to loose connections somewhere in your data or power cables.

When the code is successfully downloaded (MicroLoad will let you know that this is the case), put the two mode switches into run position – apart from each other – and press the reset switch button. An on-board red LED that is connected to Port A pin 6 will be flashing at one cycle per second.

Congratulations! You now have a stand-alone computer that performs a simple function. Inspect the code and change the software delay loop constant "#25000" to "#12500". How does this affect the LED blinking rate? Demonstrate the new flashing rate to your T.A.

Experiment 1 - Appendix: Flashing LED 68HC11 assembly language code

```
; University of Illinois at Chicago, Dept. of Electrical and Computer Engineering
; ECE 367 - Microprocessor-Based Design
; Flashing LED program for the MicroStamp11 starter kit
; V. Goncharoff and D. Yugsi, January 2006.


; Define symbolic constants

Regbas  EQU     $0000           ; Register block starts at $0000
PortA   EQU     $00             ; PortA Address (relative to Regbas)
Config  EQU     $3F             ; Configuration control register

; Begin code

; Initialize the 68HC11:

        ORG     $FF00           ; Place code in EEPROM starting at $FF00

Start:  LDS     #$00FF          ; Initialize stack pointer
        LDX     #Regbas         ; Initialize register base address ptr.
        LDAA    #$04
        STAA    Config,X        ; Disable "COP" watchdog timer


        LDAA    #$FF
Loop:   STAA    PortA,X         ; Initialize output lines of PORT A to 1's
        EORA    #$FF            ; Toggles PortA values
        BSR     Delay
        JMP     Loop


; Subroutine to achieve approximately 1/4 delay (with 2 MHz system clock)

Delay:  LDY     #25000          ; 1/2 sec = 0.5us/cycle x 25000 x 40 cycles

A0:     BRN     A0              ; Waste 3 clock cycles \
        BRN     A0              ; Waste 3 clock cycles  |
        BRN     A0              ; Waste 3 clock cycles  |
        BRN     A0              ; Waste 3 clock cycles  |
        BRN     A0              ; Waste 3 clock cycles  |
        BRN     A0              ; Waste 3 clock cycles  |
        BRN     A0              ; Waste 3 clock cycles  } 40 cycles total
        BRN     A0              ; Waste 3 clock cycles  |
        BRN     A0              ; Waste 3 clock cycles  |
        BRN     A0              ; Waste 3 clock cycles  |
        BRN     A0              ; Waste 3 clock cycles  |
         DEY                    ; 4 clock cycles        |
         BNE    A0              ; 3 clock cycles       /
         RTS

; End of code

; Define Power-On Reset Interrupt Vector

        ORG     $FFFE           ; $FFFE, $FFFF = Power-On Reset Int. Vector Location
        FDB     Start           ; Specify instruction to execute on power up
```