MPLAB8 and Flow Charts

ECE 376 Embedded Systems

Jake Glower - Lecture #3

Please visit Bison Academy for corresponding lecture notes, homework sets, and solutions

Boot Loaders

Programming a PIC Chip

- Option 1: External Programmer such as PICStart-Plus
- Option 2: Boot Loader
- Boot Loader
 - Program on the PIC Chip
 - Located at 0x000 to 0x799
 - Code must be offset by 0x800
 - Watches the serial port
 - If it sees a carriage return within 3 seconds of reset
 - It clears out the old program
 - It waits for a new program to be send via the serial port



Assembler in MPLAB8

Step 1. Create a new directory.

- Located on thumb drive works well
- X:\ECE376\ASM\Count
- Step 2. Start MPLAB8
- Step 3. Click on File New Project
- Project Wizard if this is a new project

This takes you through the process of starting a new project (i.e. a new program). Click OK Device = PIC18F4620 (next)





Program Language is MPASM

Directory for Files: Select your directory

| Project Wizard |
|--|
| Step Two: Select a language toolsuite |
| Active Toolsuite: Microchip MPASM Toolsuite |
| Toolsuite Contents |
| MPASM Assembler (mpasmwin.exe) MPLINK Object Linker (mplink.exe) MPLIB Librarian (mplib.exe) |
| Location |
| C:\Program Files\Microchip\MPASM Suite\MPASMWIN.exe Browse |
| Help! My Suite Isn't Listed! |
| < Back Next > Cancel Help |

| Project Wizard | |
|---|--------|
| Step Three: Create a new project, or reconfigure the active project? | ß |
| Create New Project File C:\ECE376_18F4620\ASM\Blink\Blink | Browse |

Click on View Project



You should see the following:



Change the default to decimal. Click on Project Build Options Project



Click on MPASM and select Decimal. This results in numbers like 100 representing 100 base 10.

| Build | Options For Project "C | lock.mcp" | | ? | × |
|--|------------------------|--------------|-------------|---------------|---|
| | Directories | Custom Build | 1 | Trace | |
| | MPASM/C17/C18 Suite | MPASM Asser | mbler | MPLINK Linker | |
| | Categories: Ge | neral | • | | |
| | Generate Command Line- | | | | |
| | | | - Default f | Radix | |
| | 🔲 Disable case s | ensitivity | C Hex | adecimal | |
| (Ext. mode now on 'suite' tab) O Octal | | | | | |
| | Macro Definitions | | | | |

The source file is what you compile.

- If this is blank, right click on Source File and select the ASM file you wish to compile.
- If you don't have an ASM file yet, select File New edit a file, and save it as .ASM

To compile your code, click on Project Bulid All (or hit key F10)

If your program compiles correctly, you get the message 'Succeed'



| Debug build of project `C:\ECE376_18F4620\A Preprocessor symbol `DEBUG' is defined. Wed Aug 20 10:14:30 2014 | SM\Blink\Blink.mcp' started. |
|---|-------------------------------------|
| Make: The target "C\ECE376_18F4620\ASM\E Executing: "C\Program Files\Microchip\MPAS Loaded C\ECE376_18F4620\ASM\Blink\Blink | |
| Debug build of project `C\ECE376_18F4620\A Preprocessor symbol `DEBUG' is defined. Wed Aug 20 10:14:31 2014 | _ SM\Blink\Blink.mcp' succeeded. |
| | - |

If there is an error in your code (such as a space in line 13 below), you will get an error message along with a notice which line has a problem

| 🜄 Jake - MPLAB IDE v8.10 | | | | | |
|--|--|--|--|--|--|
| <u>File Edit View Project Debugger Programmer Tools</u> | ls <u>C</u> onfigure <u>W</u> indow <u>H</u> elp | | | | |
| 🗅 😅 🖬 🐰 🐂 📾 🎒 🚧 🚧 🍞] 🗖 | Debug 💽 💣 🚘 🔛 🧠 🌖 🕸 🔠 🗐 📔 Checksum: 0×035a | | | | |
| Output | C:\ECE376_18F4620\ASM\CountC.asm | | | | |
| Build Version Control Find in Files Debug build of project `C.\ECE376 Preprocessor symbol `DEBUG' Pri Jan 22 12:35:45 2016 Make: The target "C.\ECE376_18F4 Make: The target "C.\ECE376_18F4620' Warning[205] C.\ECE376_18F4620' Warning[207] C.\ECE376_18F4620' Warning[207] C.\ECE376_18F4620' Error[122] C.\ECE376_18F4620' Build on first failure as reque Debug build of project `C.\ECE376_ Preprocessor symbol `DEBUG' Fri Jan 22 12:35:46 2016 BUILD FAILED BUILD FAILED | <pre>1 #include <p18f4620.inc> 2 ; Start of code: 3 org 0x800 4 clrf TRISC 5 clrf PORTC 6 mov lw 0x0F 7 movwf ADCON1 8 Loop: 9 incf PORTC,F 10 goto Loop 11 ehd</p18f4620.inc></pre> | | | | |
| | | | | | |

If you want to see what your program looks like, click on View Program Memory

| | 📉 Jake - MPLAB IDE v8.10 - Program Memory | | | | | | | | |
|-----|--|---|-------------------|----------|----------|---------|----------|----------|-----------------------|
| Fil | File Edit View Project Debugger Programmer Tools Configure Window Help | | | | | | | | |
| | D 🖻 | Project Output | 🐸 🦹 📄 Debug 💽 📑 🗃 | 🔛 🖏 | 0 🕸 | 🕮 🗈 📋 | Checksum | : 0×fb31 | |
| | 🗖 C:\E | Toolbars 🕨 | asm | 📑 Progra | am Memor | У | | | |
| | 1 | CPI I Registers | .8f4620.inc> | | Line | Address | Opcode | Label | Disassembly |
| | 2 | Call Stack | :ode: | | 1020 | 07F6 | FFFF | | NOP |
| | 3 | Disassembly Listing | | | 1021 | 07F8 | FFFF | | NOP |
| | 4 | EEDD OM | inc. | | 1022 | 07FA | FFFF | | NOP |
| | 4 | EEPROM | | | 1023 | 07FC | FFFF | | NOP |
| | 5 | File Registers | KTC . | | 1024 | 07FE | FFFF | | NOP |
| | 6 | Flash Data | OF | | 1025 | 0800 | 6A94 | | CLRF TRISC, ACCESS |
| | 7 | Hardware Stack | DCON1 | | 1026 | 0802 | 6A82 | | CLRF PORTC, ACCESS |
| | 8 | LCD Pixel | | | 1027 | 0804 | OEOF | | MOVLW Oxf |
| | | Locals | | | 1028 | 0806 | 6EC1 | | MOVWF ADCON1, ACCESS |
| | 9 | Memory | CTC, F | | 1029 | 0808 | 2A82 | Loop | INCF PORTC, F, ACCESS |
| | 10 | Program Memory | p | | 1030 | 080A | EF04 | • | GOTO Loop |
| | 11 | SFR. / Peripherals | | | 1031 | 080C | F004 | | NOP |
| | | Special Function Registers | | | 1032 | 080E | FFFF | | NOP |
| | | Watch | | | 1033 | 0810 | FFFF | | NOP |
| | | | - | | 1034 | 0812 | FFFF | | NOP |
| | | 1 Memory Usage Gauge | | | 1035 | 0814 | FFFF | | NOP |

The program is stored in the file .HEX This is a text files that contains the program in machine language (the OP-Code above)

| 4 | 102071 | |
|---|------------|---|
| | C:\ECE376_ | _18F4620\ASM\Blink\Blink.HEX |
| 1 | 1 | :02000040000FA |
| 1 | 2 | :10030000926A936A946A956A966A150EC16E822AF9 |
| I | 3 | :0403100087EF01F082 |
| | 4 | :0000001FF |
| 1 | 5 | |
| Ш | | |

To download your code to your PIC board,

- Power up your PIC board (i.e. plug it in)
- Connect the serial cable to a PC
- Run a terminal program, such as Hyperterminal or PIC_Flash_Tool
- Select the USB Serial Port (COM number varies)
- Select the .hex file to download (must be lower case letters)
- Hit RESET on your PIC board.
- Wait for Program Micro to light up
- Click on Press Program Micro.

| 🛎 GUI of super amazing awesomeness Rev | 1.2 | |
|---|---|---|
| File Instructions | | |
| COM Ports Ports Available Refresh Communications Port (CDM1) USB Serial Port (COM8) Connect To Port | COM Send File Select Hex File Program Micro Lock File Path Auto Catch Reset Must Complete the following before transfer: Waiting for file to be selected Complete: Micro Reset/Cleared Programming = | COM Received Data Recieved Data rev 02.24.17 Offset=0x800 > 32: |
| Status Dialog Made by Nathan Zimmerman USB Serial Port (CDM8) Selected Connection Established File has been selected File has been selected Micro Reset Success, Code Cleared | | Data Storage Options Clear Data To Clipboard |

Flow Charts:

Graphical way to explain how a program works

- Keep it simple (less than 20 blocks), but
- Keep it informative (more than one block)

It also helps if you follow a few rules:

- Flow charts should start at the top of the page
- The program execution should move down towards the bottom of the page
- There should be a single exit point

Flow Chart Symbols



PIC I/O

The PIC18f4620 chip has 33 I/O lines split into five ports:

| | PORTA | PORTB | PORTC | PORTD | PORTE |
|------------------|-------|-------|---------------|---------------|-------|
| Pins | 27 | 3340 | 1518, 2426 | 1922, 2730 | 3 |
| Binary Input | 5 | 8 | 8 | 8 | 3 |
| Binary Output | 5 | 8 | 8 | 8 | 3 |
| Analog Input | 5 | 5 | - | - | 3 |



Setting Up I/O Ports for Binary I/O

Three registers are associated with each port

- PORTx: Defines whether the pin is 0V(0) or 5V(1)
- TRISx: Defines whether the pin is input (1) or output (0)
- LATx: "Read-modify-write operations on the LATC register read and write the latched output value for PORTC."

In addition, you need to initialize ADCON1 to 15. This sets all I/O pins to binary.

movlw 0x0F movwf ADCON1



Random Number Generator

| #include <p18f4620.inc> DIE EQU 0</p18f4620.inc> | RB0 | | | | | |
|--|-----|----------------------|---------------|----------------------|---------------|--|
| org 0x800 | | Count really fast | Display Count | Count really fast | Display Count | |
| clrf TRISA | | | | | | |
| movlw 0xFF | | | | | | |
| movwf TRISB | | | | Start | | |
| clrf TRISC | | | | | | |
| clrf TRISD | | | Init: | | | |
| clrf TRISE | | | | nitialize I/O | | |
| movlw 0x0F | | | | | | |
| movwf ADCON1 | | | | | | |
| Main: | | | 20 | \frown | | |
| btfsc PORTB.0 | | | | RB0=1? | | |
| incf DIE,W | | | | \checkmark | | |
| andlw 0x07 | | | Roll: | yes | | |
| movwf DIE | | | K | eep rolling | | |
| movwf PORTC | | | | the die | | |
| goto Main | | | Displa | | | |
| - | | | Di | isplay Die | | |
| | | | | Value | | |
| | | | | | | |

Top Down Programming:

#include <pl8f4620.inc>
; Variables
DIE EQU 0
; ---- Main Routine ---org 0x800
call Init
Main:
btfsc PORTB,0
call Roll
call Display
goto Main



; --- Subroutines ---

Init:

| clrf | TRISA |
|-------|--------|
| movlw | 0xFF |
| movwf | TRISB |
| clrf | TRISC |
| clrf | TRISD |
| clrf | TRISE |
| movlw | 0x0F |
| movwf | ADCON1 |
| retur | n |

Roll:

incf DIE,W
andlw 0x07
movwf DIE
return

Display:

movf DIE,W movwf PORTC return

end

