CSCI 3511 Hardware/Software Interfaces

Final Project Requirements and Design

PmodCLP Project

Michael Dewar
James Brayton
Andrew Hoyle
David Chau

Fall, 2012

Project Purpose

The purpose of this project is to demonstrate the LCD PmodCLP on the Robotic Starter Kit, using the Cerebot 32MX4 PIC32MX460F512L microprocessor. The main portion of the

This project demonstrates our:

- ability to identify, formulate, and solve engineering problems;
- ability to communicate effectively;
- ability to use the techniques, skills, and modern engineering tools necessary for engineering practice;
- ability to design software and/or hardware to meet desired needs.

Project Requirements

1) The robot shall send characters to the LCD, specifically used for user information.
2) Display percent of battery life at start up.
3) Display how to use metronome on LCD screen.
4) Indicate that button 2 has been pressed by blinking the led 2 light.
5) Indicate BPM based off of timing from button 2 and button 1with both LED blinks and LCD BPM number.
6) Display timeout if user has taken too long between button 2 and button 1 presses with both LCD screen and all 4 LEDs.

Project Design

The program uses various interrupt (ISR) routines to manage hardware input, output, and timers.

Program Files

File	Description
main.c	Main program file.
#include	<plib.h>, "stdtypes.h", "config.h", "MtrCtrl.h", "LCD.h", <stdio.h> <string.h>

Custom Parameters

Parameter	Description
TIME_FACTOR	Use as parameter in delayUs() to control the delay time
stPressed	Use to keep track and control the state of button 1 and button 2
stReleased	Use to keep track and control the state of button 1 and button 2
cstMaxCnt	number of consecutive reads required for the state of a button to be updated

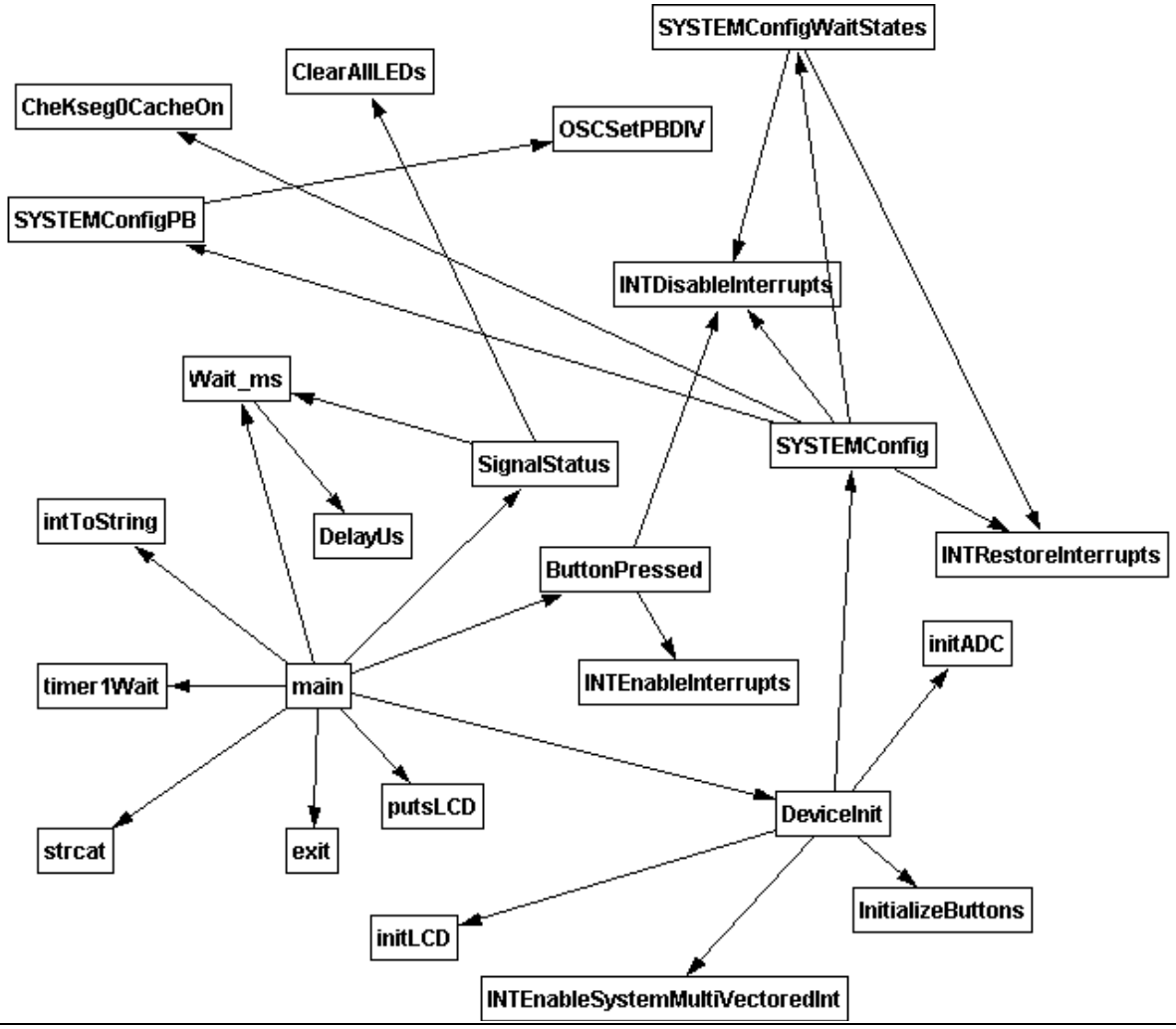
Main Public Interface (forward declaration in the main.c)

Function	Description
initADC()	Initialize ADC
initLCD()	Initialize LCD
readADC()	Read data from channel 8
DeviceInit()	Initializes on chip peripheral devices to the default state
Wait_ms()	Will make motors wait for specified number of milliseconds
void DelayUs(WORD tusDelay);	This procedure delays program execution for the specified number of microseconds.
intToString()	Convert string of numbers to integer
cmdLCD()	Use to clear content on LCD or use to position text display on LCD
putsLCD()	Send content to LCD for displaying
myWaitMs()	Will wait for specified number of milliseconds
ButtonPressed()	Use interrupts for button 1 and button 2
ButtonPressed2()	Use to keep track of two buttons status
ClearAllLEDs()	Clear all leds
drawProgressBar()	Draw progress bar
Delayms()	Delay the Timer 1 for a number of milliseconds
clrLCD()	Clear contents on LCD
void InitializeButtons()	Initializes the state for buttons 1and 2.
BOOL SignalStatus(WORD status)	Displays the proper LED sequence given which WORD parameter status is passed in.

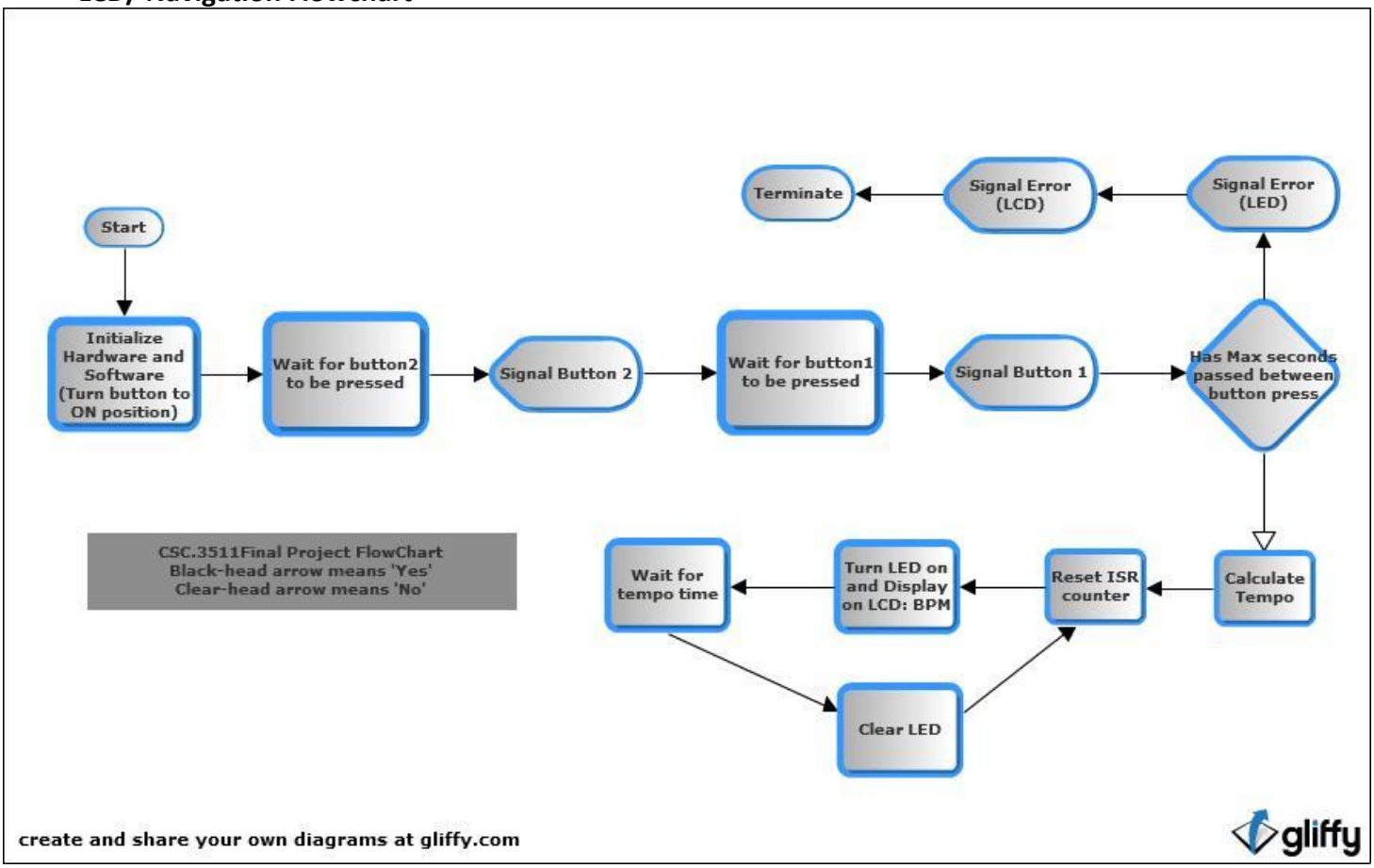
Event Interface

ISR	Description
mT5ClearIntFlag()	Clear interrupt flag for Timer 5
INTDisableInterrupts()	Disable interrupt service
INTEnableInterrupts()	Enable interrupt service
INTEnableSystemMultiVectoredInt()	Enable multi-vector interrupt services

Function Call Graph


SYSTEMConfigWaitStates
ClearAllLEDs
CheKseg0CacheOn
OSCSetPBDIV
SYSTEMConfigPB
INTDisableInterrupts
Wait_ms
SYSTEMConfig
SignalStatus
intToString
DelayUs
INTRestoreInterrupts
ButtonPressed
initADC
timer1Wait
main
INTEnableInterrupts
putsLCD
DeviceInit
strcat
exit
InitializeButtons
initLCD
INTEnableSystemMultiVectoredInt

LCD/ Navigation Flowchart



Configuration / Execution Instructions

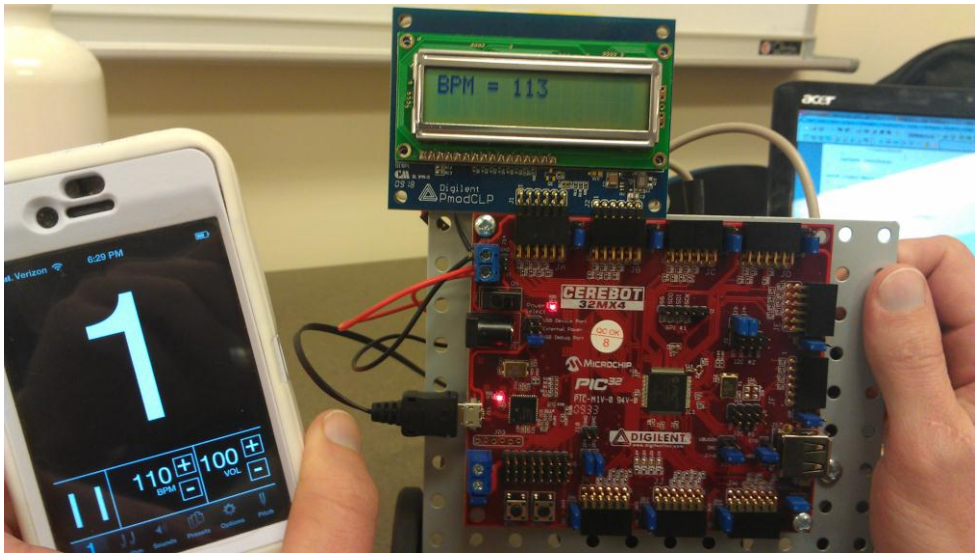
Software Development Environment

MPLAB IDE v8.76 embedded with C32 C compiler (pic32-gcc.exe) v.12

Hardware Configuration Bit Settings

Bit	Value	Description
FPLLIDIV	DIV_2	PLL Input Divider
FPLLMUL	MUL_18	For spi
FPLLODIV	DIV_1	PLL Output Divider
BWP	OFF	Boot Flash Write Protect
CP	OFF	Code Protect
FNOSC	PRIPLL	Oscillator Selection
POSCMOD	HS	Primary Oscillator
FWDTEN	OFF	Watchdog Timer

Robot Setup and Pre-Conditions



Results

1. **The robot will act as a metronome and display battery life**
   The robot upon startup will display the battery life. After a constant amount of time, the robot will then display a "Menu" for the user to choose from. The first program will go into the metronome program while the second program will display the members on our team.
2. **Proper information will be sent to the LCD**
   If the Metronome program is called, then the user will have a specific amount of time to press the buttons used for the tempo of the metronome. If a button is not pressed within the time frame, then the LCD will display "You ran out of time"; otherwise the program will display on the LCD the BPM's.
3. **LED's will pulse based on the metronome tempo**
   The metronome program will pulse the LED based on the interval given by the user through the button presses.

For the most part, we have made the robot pulse and act like a metronome and display it to the LCD.

Error Control

1. The robot does calculate a specific metronome but due to human error in how accurate one can press the button, the program is too efficient and takes that human error and computes the tempo based off of that. For the most part, the robot could be used for a short period of time but if used for a prolonged amount of time, the inaccuracy within each iteration would increase leading to a larger and larger margin of error.

2. Logic errors in the program included simple tasks such as displaying the LED for a specific number of time before clearing it, otherwise it would display it but without our knowledge. The same concept could be applied to the LCD as well in that we needed to display something on the screen long enough for use to see the display before clearing

3. We also noticed when the battery is weak, the LCD still displays content but the it could be vague and lightened compared to the regular display with regular power. When this happens, battery needs to be replaced.

References

Document Name	Revision	Description
Digilent Robotic Starter Kit with Cerebot 32MX4 Reference Manual	10/8/2009	Describes how to assemble the robot.
RSK_Cerebot32MX4_rm.pdf	8/26/2011	Describes how to set up robot
http://benrobotics.wordpress.com/2010/06/10/my-first-robot-part-iii/	11/21/2011	Describes how to set up and capture data to calculate the battery life
http://digilentinc.com/Data/Products/CEREBOT32MX4/Cerebot_32MX4_rm.pdf		Describes how to set up and capture data to calculate the battery life
Programming_PIC32_in_C_Lucio Di Jassio.pdf		Describe how to initialize ADC and capture data from channel 8 for calculating percent of battery life
Startup project	10/20/2012	Used as startup for this project
Navigation project	10/20/2012	Used as startup for this project
Simon Says Project	11/28/2012	Used for startup and reference for this project

EXAMPLE CODE: emailed to Gnabasik