Bachelor of Engineering in Electronic Engineering

(Level 8)

Project Report

Bike Computer

14TH December 2018

Supervisor: **Tom Murray**

Submitted By: **Carl Sagrado**

Student No: **X00084403**

Declaration

I hereby certify that the material, which I now submit is entirely my own work and has not been taken from the work of others except to the extent that such work has been cited and acknowledged within the text of my own work.

Signature: ____Carl Sagrado_______________ Date: ___14/12/2018_____

Contents

Abstract

Bike computer is an electronic device that is used in monitoring the speed of the bicycle as it travels over a distance. This device is perfect for tracking how fast and how long a bike user has been on the road. A must-have device for cyclist that allows you to determine the speed through the LCD display which can be a fantastic motivation tool when working out. This device can also serve as a reminder in regard to safety for cyclist on the road to remind themselves of the speed they are going at in order to avoid road accidents. Fitted navigation button to navigate through the whole system, users can manually adjust features such as brightness control, contrast level display, timer, etc. With brightness control, users can control how bright they want the device to illuminate which is perfectly suited for those who prefer cycling at night. A contrast control can manually be adjusted to preferred level when the default display is not ideal for the user. To which the whole operation of the bike computer is processed by a single microcontroller called, MSP430. This device is a low powered device fitted with voltage regulators which can guarantee that the circuit will run smoothly when fitted with a 9V battery. Inside this document entails the full description used components used and its operation.

Acknowledgements

I wish to thank my project supervisor Tom Murray for the unlimited guidance, information, and materials throughout the development of this project. His insight and guidance to the project brought enormous influence for the completion of the project. I would also like to thank Pat O' Friel for the additional information he has provided me throughout. I am particularly thankful to our college technical officer, David Maguire for the resources he has given me when I needed them the most. A special thank you to Bill Black and Hugh Brennan for the encouragements and support throughout the years, thank you both for being understanding and supportive through this academic journey. Finally, I am grateful to colleagues and students at Institute of Technology Tallaght, from whom I've received enormous support during the academic as well as fun and memorable experience while studying at this institute.

Introduction

The aim of this report is to provide details of the planning, design and construction of Bike computer project. Bike Computer is an electronic device that can be used in monitoring the speed of the bicycle as it travels over distance. The concept of the project is to use a microcontroller to read the rotation of the wheel and takes in the data to be calculated relative to the radius of wheel, which will then be displayed in an LCD display. Navigation buttons are available to provide users ease and control over the device to navigate the features available such as brightness adjust, elapse timer, contrast level and a system reset. To simulate a wheel rotation, a function generator is used to send a signal to the circuit which is read by the microcontroller and by using an interrupt function, we were able to capture to the full rotation of the wheel by attaching an instruction to be executed as the signal rises.

The first chapter of the report entails the power supply unit that will be used in the project. Since we aim to achieve a low powered bike computer, we assume that we require 3.3V and 5V to power components for the bike. Mainly, The LCD display would require 5 V to power up and the rest of the switches, sensors and microcontroller would require 3.3V to function.

The second chapter provides the details of the component display that we are going to use to display information which will be seen by the user. Alphanumeric 16x2 LCD is an electronic display module used in this project to display visual results of our bike computer. Since, most of the functionalities and calculation are embedded into the microcontroller, this module is used to display the relevant information to the user. Within this chapter contains the theoretical operation of the display with a full description of the component.

The third chapter contains the main brain of the bike computer, MSP430. MSP430 is a microcontroller that takes in sets of instructions to which will execute commands at a pre-defined condition. The MSP430 demands very low power, 3.3V, which is economically friendly and very reliable when executing well defined tasks. More information about the microcontroller can be seen in the chapter.

The switches and sensors used in this project will be documented in this chapter which will entail all the work arounds of the switches. The menu and navigation switches are provided to allow users navigate the whole with ease as it provides an interactive process which are all displayed on the LCD. The calculation of the bike computer (speed calculation) is detailed in this chapter following an explanation to the simulation activity to which we simulated the wheel rotation using a square wave signal.

The results and comments of the testing is available in this document to provide a thorough documentation of how the components work and the whole project as a whole.

A user manual for the operation of the bike computer is available for now technical users that would enable them to utilize the device with ease and satisfaction.

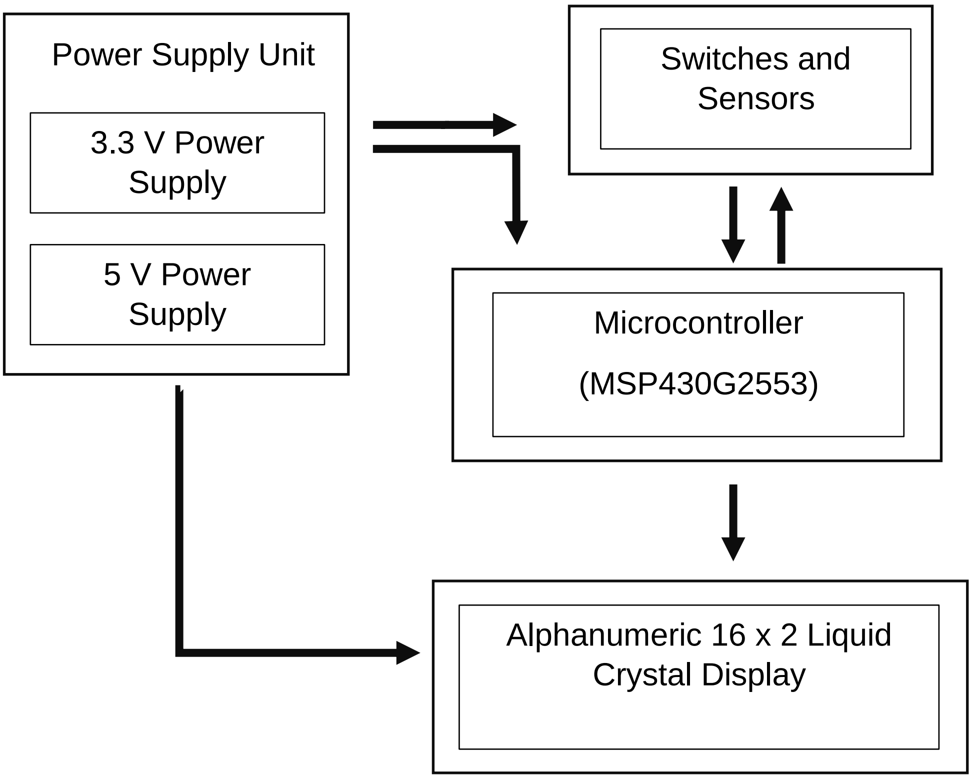
Project Planning

The table below shown in Table 1, shows the plan laid out for the duration of the semester. The project will take thirteen weeks to complete and expected to produce a working prototype which will be demonstrated at the end of the semester. For each week there are tasks that are set to complete with addition to new components to the project as the project progresses.

Table 1: Project Plan

Week	Date	Description
1	21/09/18	Project Selection and Soldering
2	28/09/18	5 V Power Supply Construction and Testing
3	05/10/18	3.3 V Power Supply Construction and Testing
4	12/10/18	LCD Display - Manual testing - Circuit construction
5	19/10/18	
6	26/10/18	
7	02/11/18	LCD Display and MSP430
8	09/11/18	Switches and Sensors
9	16/11/18	Software and testing - LCD and MSP430 testing - Switches Testing - Sensor Testing - Added features testing
10	23/11/18	
11	30/11/18	
12	07/12/18	Quiz (1-hour max) – 15%
13	14/12/18	Hardware demo / Logbook/Report

Overview of the Circuit



Full System Block Diagram of the Bike Computer

Theoretical Operation

DC Power Supply

The primary function of a power supply in our project is to covert electric current from any source to correct voltage and current. It will regulate voltage to maintain constant output voltage and avoid further damage to our components. DC Power Supply is one of the core fundamentals in having a successful product. As mentioned in the planning stage, we aim to use an Alphanumeric 16x2 LCD and a microcontroller to which requires 5v and 3.3v, respectively. We will go through each component used in the next sections of this report and discuss their functionalities and operation and how paramount a correct supply of power to these components.

As a unit, power supply is divided into two parts which outputs 5 V and 3.3 V as seen in Figure 1.

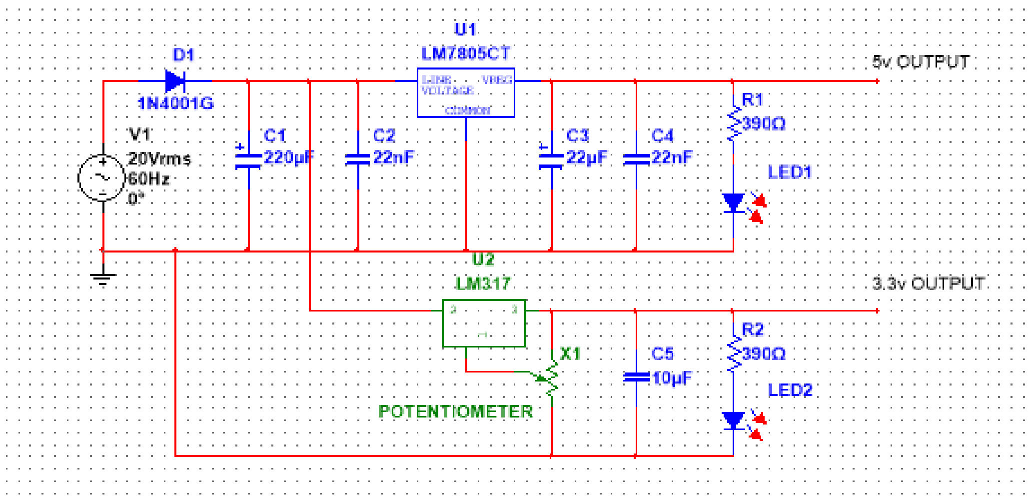


Figure 1: Power Supply Unit

Diode

Before we proceed to discuss the two voltage regulators in the power supply, it is important to take note of one important component, called diode, which sits passively at the input of our power supply connected in series with the input of each regulators, as seen in Figure 1. The main function of the diode is to allow an electric current to flow through it in only one direction, creating a constant one directional power flow to our regulators. As such, this diode can also be viewed as a check valve that is useful to convert alternating current (AC) to direct current (DC). The diode used in the circuit is silicon diode which is fixed by the chemical composition of the P-N junction and have a voltage drop approximately 0.7 volts. When a diode is placed in forward-based (positive to negative) this acts as a rectifier, as previously stated has voltage drop of 0.7 volts. However, when placed in reverse (reversed-based) would block current flowing through the circuit. Orientation is shown in the figure 2 below.

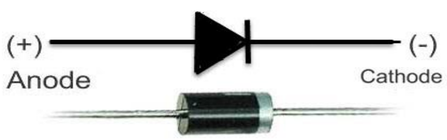


Figure 2: Diode Orientation [12]

5.0 V Power Supply

To supply constant fixed output voltage into the circuit, a voltage regulator is required to be applied into our circuit. As mentioned, a voltage regulator maintains the output voltage at a constant voltage, as such a voltage regulator, LM7805, a member of 78xx series of fixed linear voltage regulators is used to maintain fluctuations and excess of undesired voltage supply.

LM7805 is an orientation dependent IC, as observed in figure 2 and is operational at temperature up to 120°C. The IC TO-220 package can take a voltage input ranging from 7v up to 35v with output voltage range with the minimum of 4.8v up to 5.2v max and can deliver up to 1.5A of output current.

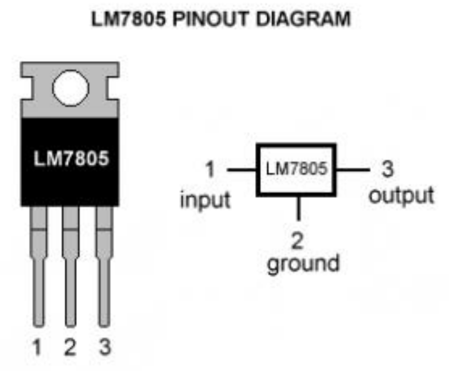


Figure 3: LM7805 [[16]]

As noticed, the significant difference between the input voltage and output voltage is released as heat. Thus – the greater the difference between the input voltage and the output voltage, more the heat generated. This is important to take into consideration when applying power into the circuit as more heat dissipated could potentially damage the regulator or cause it to malfunction. It is advisable to apply input voltage to a maximum of 5v above the output voltage.

Several capacitors are applied to the input pin and output pin of the LM7805 which will act as a smoothing component, they are in place to provide stabilized voltage input and output voltage. A raw DC supplied by the diode would consist of a series of half sine waves with voltage varying between zero and root two times the RMS voltage, a suppled power of such would not be sufficient when powering digital circuit would not work because the power would be removed every half cycle.

An LED in series with a current limiting resistor connected at the pinout of the regulator, this is designed as such to act as an indicator for the 5v supply.

3.3V Power Supply

The regulator used to supply a 3.3v output is variable voltage regulator. LM317 is a 3-terminal linear voltage regulator that is fully adjustable to which its output can be varied by using the ratio of two resistances*, (to which can also be achieved by using a variable resistance).* [1]The output voltage can be set to the desired level with a corresponding input voltage being anywhere between 3v to 40 v. The adjustable voltage regulator IC LM317 develops 1.25v nominal reference voltage (V_{ref}) between its output and the adjustable terminal. When desiring a constant current flow through one of the output pin, the output voltage can be calculated as follows: $V_{out} = V_{ref}\ (1 + \frac{R_2}{R_1})$. The output voltage of the regulator ranges from 1.25v to 37v. LM317 is an orientation dependent and the pin out is shown in figure 3. The IC is operational at up to 120°C and capable of supplying more than 1.5 A over an output-voltage range of 1.25 V to 37 V. Embedded into the internals of the regulator, it contains a short-circuit current limiting, thermal overload protection and safe operating area protection [4].

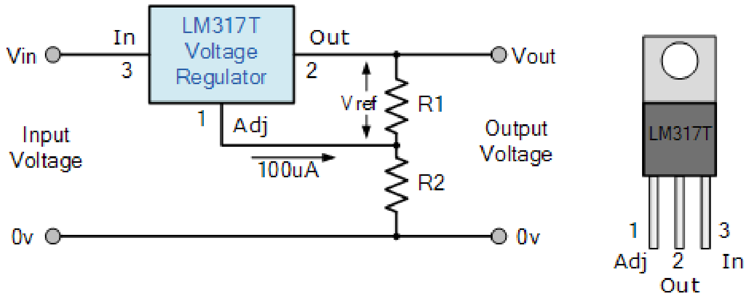


Figure 4: LM317 - 3.3 V Supply [17]

For the prototype, instead of using two resistors and work out the resistance ratio for the output voltage, we used a 10k Ω variable resistor to provide a ratio for the output voltage. With this, the power supply would be scalable and usable to provide desired voltage in the future.

Another smoothing capacitor can be observed in the 3.3 V output, seen in figure 1, to smoothen any unwanted noise and transient signal coming from the output.

A similar set up to the 5 V supply where an LED in series with a current limiting resistor is applied to the output pin of the voltage regulator is used to indicate the functional operation of the output voltage. In doing so, a malfunction on the voltage regulator, for example, no output voltage or too much of output voltage coming out from the regulator, would render the LED off. This setup allows a visible troubleshooting method in the event of the prototype not working or malfunction.

Alphanumeric 16x2 LCD

Alphanumeric 16x2 LCD is an electronic display module used in this project to display visual results of our bike computer. Since, most of the functionalities and calculation are embedded into the microcontroller, this module is used to display the relevant information to the user.

16 x 2 is a type of display which can be translated to a display of 16 characters per line in 2 such lines, as indicated in figure 5. This indicates *(16 x 2 = 32*) 32 characters in total can be displayed and with each character displayed in 5x8 pixel matrix (5 x 8 dots with cursor), as shown in figure 5 indicated by the arrow.

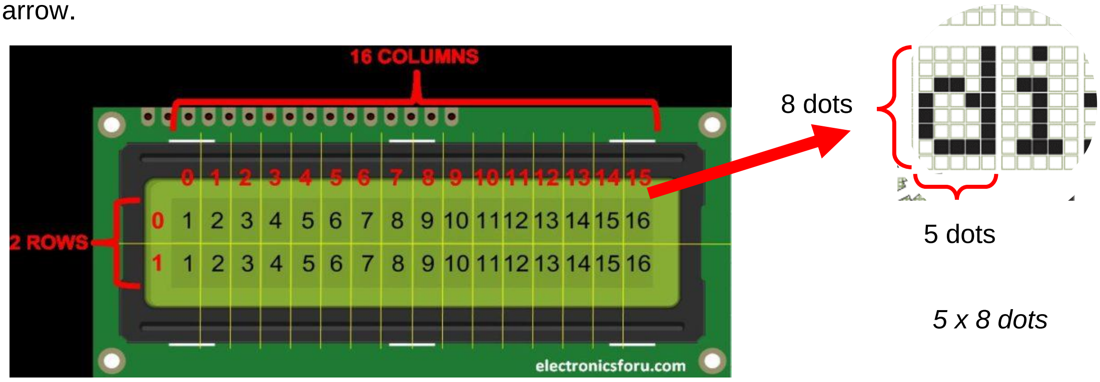


Figure 5: LCD 16 x 2 Characters [[18]]

The 16x 2 LCD Module is operational from 4.7 V up to 5.3 V with current consumption is 1mA without the use of the backlight. LCD display have a controller chip on board, typically an Interface IC like Hitachi HD44780 or equivalent, to take commands and data from the MCU and process them to display information onto our LCD screen. The pin configuration is illustrated and described below in figure 6 and Table 1, respectively.

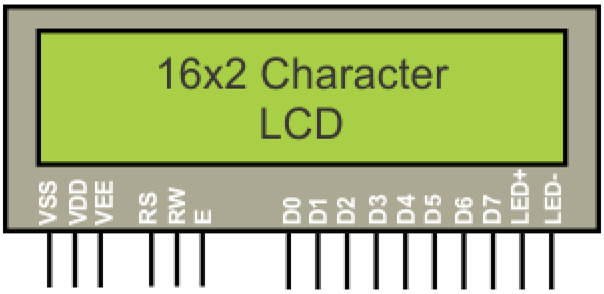


Figure 6: 16 x 2 LCD Pin Configuration [2]

Pin	Symbol	Function/Description	
1	VSS	Ground	
2	VDD	+5 V (Main power supply)	
3	VEE	Contrast Adjust – (The contrast can be adjusted through this pin, simply by providing a variable resistor through Vcc.)	
4	RS	Register Select (0 = to select command register, 1 = to select data register)	
5	RW	Read/Write signal (0 = to write to the register, 1 = to read from the register)	
6	E	Enable Latch data or commands into the module after low to high transition.	
7	D0	Data bit 0	Display Data Signal
8	D1	Data bit 1	
9	D2	Data bit 2	
10	D3	Data bit 3	
11	D4	Data bit 4	
12	D5	Data bit 5	
13	D6	Data bit 6	
14	D7	Data bit 7	
15	LED+	+4.2V for LED	
16	LED-	GND (0V)	

Table 2: Pin Description

Register Select Pin

The LCD has two registers, namely Command and Data. As indicated in Table 1, The register select pin is used to select from one register to the other. When the pin is low any data latched into the display will be interpreted as a command. A command is an instruction given to LCD to do a predefined task like initializing, clearing the screen, setting the cursor position, controlling display etc. When register select pin is high, the data register will interpret the data as character information such as “ABCD!2£” which will be stored to be displayed on the LCD.

Enable pin

All the register selection is enabled by the Enable pin. When a register is chosen, this pin sends data to data pins when a high to low pulse is given. In other words, if we want execute instruction to the LCD display, this pin has to be toggled low to high transition following a selection in the register.

Read/Write pin

Two functionalities are provided by this pin. As the name suggest, this pin is for reading or writing to the register. When the pin is low, the module will accept any information and writes to the register. Inversely, when the pin high we can read information from the register. In such case, we can use this pin to write information to the LCD or to take information out from the LCD module such as the Module Busy flag. Typically, we set the pin to ground to write data to the LCD. [3]

Contrast pin

As stated in the table 1, the pin allows adjustments of LCD contrast. A voltage varying from the supply voltage (+5 V dc) to ground (0 V) will adjust the contrast of the pixels on the display. A fixed resistor at the predefined calculated voltage value can set to a stable and fixed contrast level or by using a variable resistor can provide with a much flexible level of contrast that can be changed at any desired time.

Data bit pins (D0-D7)

Data pins D0 to D7 forms an 8-bit data line, which are the data bus connections used to carry the information into or out of the module. As these pins serves as the data bus connections, they are also used to send command or data to the LCD. Depending on the register selection, these pins can also be called Data or Command pin.

There are two types of wiring mode available when interfacing the data bit pins to an I/O port of a microcontroller or manually wiring the display, namely 4-wire mode and 8-wire mode. As the name suggest, 4 wire mode uses only 4 pins (D4 to D7) and the 8-wire mode uses 8 pins (D0 to D7). The two wiring modes are also called 4-bit interface and 8-bit interface. The difference between the two is that 4-bit data transfers will take twice as much time to transfer data as 8-bit data transfers. However, as previously stated, the 4-bit mode only requires only 4 pins to transfer data which is crucial when I/O pins are limited compared to 8-bit mode. Thus, with limited I/O pins with no time constraints, 4-bit mode is advisable. If the I/O pins are plentiful but time is relevant to the application, 8-bit mode would be the preferable method.

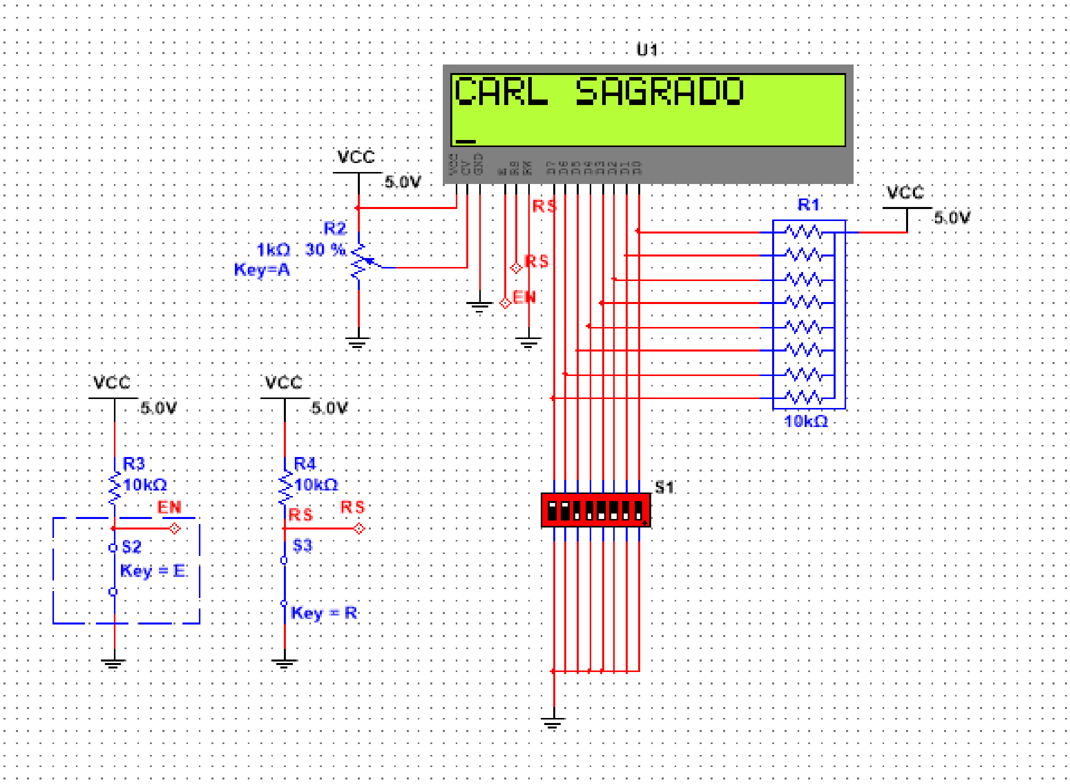


Figure 7: LCD Testing

The interface mode shown in figure 7 is an 8-bit interface mode to manually test the LCD display. The data pins are connected to an 8-way dip switch in parallel with resistor pack which will act as a current limiting resistor. This set up enables us to control the module with byte by byte transfer. By connecting the adjust pin to a variable resistor, we are able to control the contrast level of the display. As observed, Read/Write pin is directly connected to the ground as we only need write data to the LCD, as intended. The two switches connected to enable, and register select allows us to toggle between command and data as we go along the testing.

With reference to the datasheet of the LCD display, a step pattern is required during the initial power up of the display. At which on the initial power up, the default display is in off mode. This indicates that nothing will be shown into the screen. This can be turned on by setting 'display on' which requires a parameter in 8-bit binary or byte value to enable this function (e.g. 00001111), full command control codes are shown in table 2. As the binary value is loaded as a command, the register select (RS) pin has to be set to low while enable pin is set from low to high. The result, LCD should display a cursor that blinks. Next, we need to set the LCD to operate in 2-line mode. To enable this function, we follow the command code set in table 2 "function set" and input the required binary value using the data pins (e.g. 00101111) and toggle, RS to low while toggling enable pin from low to high to send the command to the display. To set the LCD to display characters from this point, is done by simply entering the binary value of each character to the data pins, and toggle enable button. Note as previously stated in table 1, to display characters in the LCD, register select must be high while toggling enable pin from low to high.

Table 3: LCD Command Control Codes [4]

Command	D7	D6	D5	Binary D4	D3	D2	D1	D0	Hex
Clear Display	0	0	0	0	0	0	0	1	01
Display & Cursor Home	0	0	0	0	0	0	1	x	02or 03
Character Entry Mode	0	0	0	0	0	1	I/D	S	04 to 07
Display On/Off & Cursor	0	0	0	0	1	D	U	B	08 to 0F
Display/Cursor Shift	0	0	0	1	D/C	R/L	x	x	10 to 1F
Function Set	0	0	1	8/4	2/1	10/7	x	x	20 to 3F
Set CGRAM	0	1	A	A	A	A	A	A	40 to 7F
Set Display Address	1	A	A	A	A	A	A	A	80 to FF

I/D: 1 = Increment*, 0 = Decrement
S: 1 = Display shift on, 0 = Display shift off*
D: 1 = Display On, 0 = Display Off*
U: 1 = Cursor underline on, 0 = Underline off*
B: 1 = Cursor blink on, 0 = Cursor blink off*
D/C: Display shift, 0 = Cursor move

R/L: 1 = Right shift, 0 = Left shift
8/4: bit interface*, 0 = 4 bit interface
2/1: 1 = 2 line mode, 0 = 1 line mode*
10/7: 1 = 5 x 10 dot format, 0 = 5 x 7 dot format
x = Don't care * = Initialisation settings

Microcontroller

Microcontroller in general is a self-contained system with processor, memory and other peripherals. In regard to this project, it is an IC chip that contains embedded instruction to execute instruction to control and enable specific application.

Different microcontroller consists of several devices featuring different sets of peripherals for various applications are available to purchase online however, a microcontroller chosen to be used in this project is a type of microcontroller that requires low supply-voltage that is capable of outputting mixed signal called MSP4302553 which will be discussed below. This microcontroller will serve as the brain of the whole project and the core fundamental element in communicating other elements in our prototype.

MSP430G2553

The MSP430G2553 IC, from the MSP430 family is of ultra-low-power microcontrollers. This type of microcontroller only requires voltage range from 1.8V to 3.6V to power on and has 16-bit RISC architecture with 62.5 ns instruction cycle time. This means that it is designed to be efficient at communicating with data buses in performing task at a higher speed with low power consumption. There are other alternatives to MSP430 such as Arduino Uno R3 microcontroller with similar specification as to our chosen microcontroller, however, although it provides more available pin sets it requires more power to run compared to MSP430G2553. [5]

The MSP4302553 microcontroller shown below in figure 8 has 20 pins. 16 pins of which are available to use depending on the application and 4 pins are specifically reserved and are used for supplying voltage to the IC (pin 1) and ground (pin 2). The other reserved pins, pin 17 and pin 16, are used for resetting and testing. The testing pin is used to communicates to the IC especially when writing instruction to the chip. The rest of the available pins can be used as an input or output pins depending on the application as previously mentioned. The main functionalities of the reserved pins are the ability to digitally read/write as well as able to read or write in analog. The pins for different functionalities are labelled in figure 8 to which most of which will be used when testing and implementing the bike computer project.

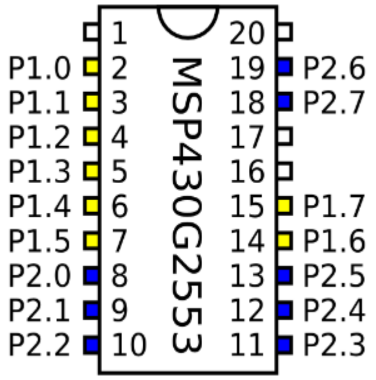


Figure 8: MSP4302553 Pin outs [19]

The integrated development environment (IDE) we are going to use to communicate to the IC is Energia. Energia IDE is an open source platform that uses the mspgcc compiler in order to communicate to the MSP430. It is a portable framework layer that can be used to other popular IDEs.

The testing of the MSP430 communicating to the LCD display is documented at the Testing chapter. A full set of Liquid Crystal library is available online that enables the MSP430 to control the LCD and are compatible with the Hitachi HD44780 driver. The code for the testing is available at the appendices page which are commented for learning purposes.

When attempting to display information to the LCD using the MSP430, similar steps are needed to be called when initialising the interface to the LCD screen, which requires specification of the intended dimensions (width and height) of the display. Before anything else, begin() needs to be called before any other LCD library commands which takes in (e.g. lcd.begin(cols, rows);).

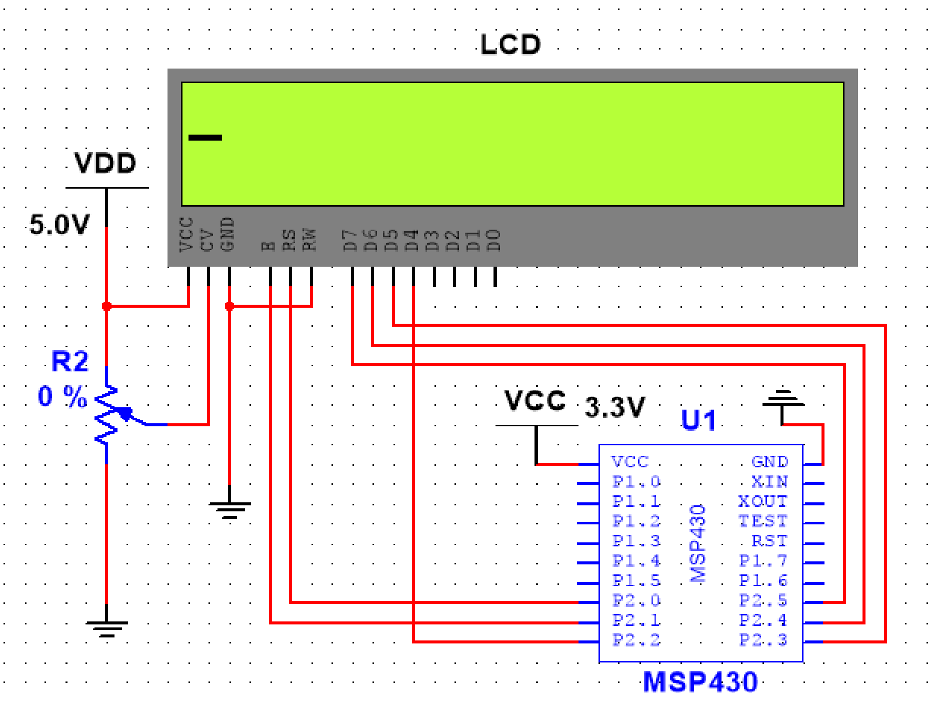


Figure 9: MSP430 and LCD

Switches and Sensors

Switches were added to the functionalities of the bike computer to provide an ease of the navigation to the bike computer. This would allow users to be more interactive to the project and at the same time provides an ease to control when sending and receiving information to the microcontroller.

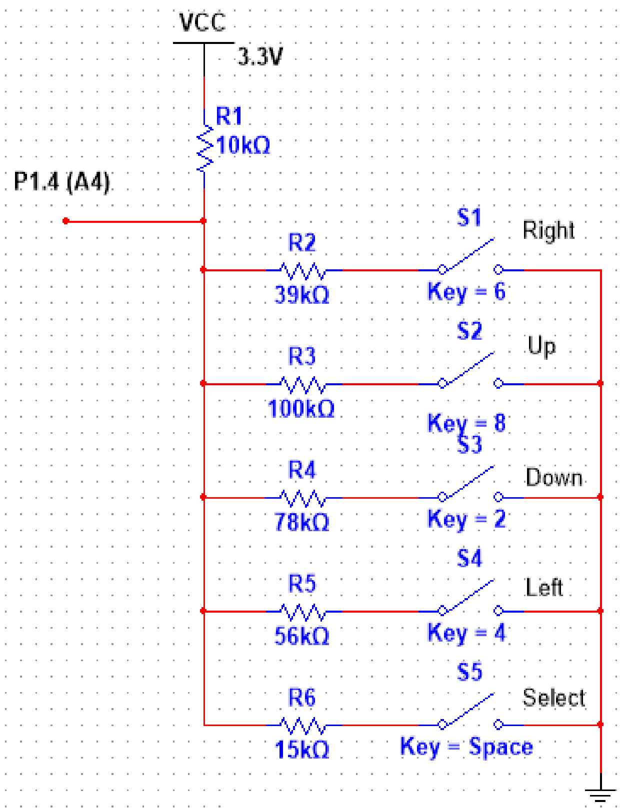


Figure 10: Switches

Switches are connected in parallel with each other where they are all feed with 3.3 V supply in an open circuit. This indicates that when all switches are all high, the 3.3 V supply is constant. The variation in voltage output of the supply is when one of the switches are triggered (set high to low) outputs a varying result from the main 3.3 V. This activity is sent and read by the analog pin of the MSP430 which are set with pre-defined conditions whenever a specific voltage is read.

For example, when S1 (Right switch) is set to low, the combined resistance of 49K Ω (10KΩ + 39KΩ) will pull down voltage from the supply which are calculated and measured to be 2.627 V. This means that at the instance the S1 is triggered to low, the reading of the microcontroller will be having 2.626 V that will meet condition on the MSP430 triggering a n event.

Full list of table below are the measured values when one of the switches are triggered. This table will be referred to of the working code available in the appendices. The table also indicates the proper switch names and its orientation to enable an easy identification of the switches. Full testing and procedure of switches is available at the testing chapter of this report.

Schematic ID	Name	Voltage Value	Function
S1	Right	2.627 V	Stop/Reset (Brightness Adjust +)
S2	Up	3. 0 V	Move menu up
S3	Down	2.925 V	Move menu down
S4	Left	2.8 V	Start/ Pause (Brightness Adjust -)
S5	Select	1.98 V	Menu select

Figure 11: Switches table

In addition to switches, to determine the rotation of the wheel a magnetic switch can be used (also called reed switch). The switch is also used to measure the speed of one's bike as it rotates. This project takes that measurement, do calculation and displays on the LCD screen. The magnetic switch works by trigger, meaning as the wheel makes a full rotation the reed switch and the magnet interact setting the value of the reed switch to max (~1023) or 0. This indicates that when the wheel has yet to make a full rotation, the reed switch value will be equal to 0. This event can be considered to be low (has yet to make 1 full cycle) or high when max value at ~1023 (at full cycle).

The calculation of the speed of the bike is defined as the distance travelled divided by time taken. Anything that rotates such as a wheel of a bike is defined by the number of revolutions carried out per minute. From there the result can then be converted between revolutions per minute and linear speed.

$$\omega = 2\,\pi\, f,\ \ since\ f = \frac{1}{T}$$

Angular velocity can also be defined by: $\omega = \frac{Circumference}{Time}$

$$\omega = \frac{2\pi}{T}\ \ (RPM)$$

Omega (ω) is mathematical notation for angular velocity, the unit is revolutions per minute (RPM). Linear speed is equal to the velocity of a moving object, the unit is metres per second (m/s).

To convert revolutions per minute (RPM) to linear speed (mps), the formula is show below.

The variable required to determine the correct linear speed in relation to the wheel of the bike is the radius. The formula is given,

$$v = \ \omega * r \qquad \text{where r = radius}$$

Therefore, the full equation to get the linear velocty (m/s) is:

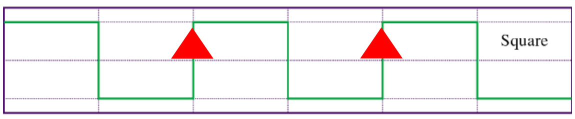
$$v = \ \omega * \ r * 2\pi * 0.016\ or\ time/distance\ \text{(m/s)}$$

To test the calculation given above, we are going to simulate the reed switch event using a function generator and an attachInterrupt function with the MSP430.

As seen in figure 12, the schematic shows that we can simulate the reed switch values per wheel rotation by taking a square wave to the P1.6 and the full rotation of the wave would be equal to one full revolution of the wheel.

By implementing a method to interrupt each rotation would provide the desired full wheel cycle as seen below.

start *stop*



\+ *1 revolution* +

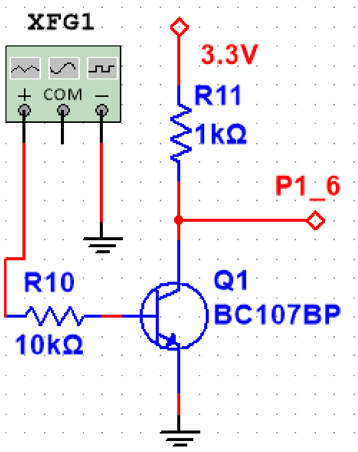


Figure 12: Reed Switch Simulation

attachInterrupt()

attachInterrupt() - is a function call that interrupts any processes the processing is executing and inserts its own instruction to executed first before anything else.

When interrupt occurs, the method inserts specific set of instructions inside the function and execute these commands to with higher precedence of order over the rest of the instruction where interrupt is triggered. For example, when the interrupt is called at the rising edge (where pin from low to high) this inserts interrupt actions which leaves the rest of the code at half until the full interrupt function is executed. The function attachInterrupt takes in a set of parameters to sucessfully work namely pin, ISR and the mode it would work on.

Parameters

Interrupt pin – The first parameter to attachInterrupt function is the interrupt pin, it requires a digital pin to translate actual digital pin to specific interrupt number.

ISR – Also known as Interrupt Service Routine, is a call when interrupt occurs. This function takes no parameters and return nothing.

Mode – this refers to the location at which the interrupt is triggered. This parameter contains predefined 4 constant values and only such is accepted, these values are low, change, rising and falling.

For this project, we are going to set the interrupt at the rising edge of the wave. As the wave rises, this will indicate the start of the cycle. When reaching the full cycle of the wave at the next rising edge, it would be taken as the the full revolution of the wheel of the bike. Important note for this project when using the simulation is to set the interupt function at the initialising stage of the microprocessor or as the microprocessor powers on.

With relevance to the bike computer, as the full cycle occurs the interrupt function will take the difference of the present time to the previous time which will be taken as the revolution per minute and further assign the current time to the previous time to be used for the cycle. This process allows us to determine the difference of speed at which the full cycle.

Pulse-Width Modulation

Pulse-Width Modulation (PWM) signal is a method for controlling analog circuits with a microprocessor. In theory, the modulation works by sending an analog signal to the circuit through a digital source such as microcontroller. The modulation has two components, duty cycle and frequency. Duty cycle refers to the amount of time for the signal stays in a high state as a percentage of the total time PWM completes to one full cycle and the frequency refers to the speed at which PWM completes a cycle. This type signal can be used in varying the controlling the brightness level of our LCD by applying PWM signal into NPN transistor. The transistor acts as a swtich between 5V supply and the anode of the LC switch which will be closed dependent to the amount of time the PWM pulse is set.

Bill of Materials

Sch. ID	Component	Part No.	Supplier	Qty.	Price
D1	IN4001	1N4001-E3/54	Radionics	1	€0.03
C1	220uF (Electrolytic)	ECA-1CM221B	Radionics	1	€0.05
C2, C4	22nF	890324023011CS	Radionics	2	€0.32
U1	L7805	LM7805CT	Radionics	1	€0.46
C3	22uF (10v)	UPW1H220MDD	Radionics	1	€0.05
U2	LM317	LM317T	Radionics	1	€0.34
C5	10uF	ECA1HM100	Radionics	1	€0.05
R1, R2	390 ohms	RS-Carbon-390R-5%-0.25W	Radionics	1	€0.11
LED1	LED (Red)	L-53SRD-12V	Radionics	1	€0.26
LED2	LED (Blue)	SSL-LX5093USBC	Radionics	1	€0.44
Pot1, Pot2	Potentiometer(10k)	TRIM-10K	Hobbytronics	2	€0.60
LCD_1	LCD 2x16	162C-BC-BC	Radionics	1	€11.74
MC	MSP430G2553	MSP-EXP430G2	Radionics	1	€10.11
R3	10K Ω	RS-Carbon-10k-5%-0.25W	Radionics	1	€0.01
R4	39K Ω	RS-Carbon-39k-5%-0.25W	Radionics	1	€0.01
R5	100K Ω	RS-Carbon-100k-5%-0.25W	Radionics	1	€0.01
R6	78K Ω	RN65D7802FB14	Electra Group	1	€0.01
R7	56K Ω	740-0918	Radionics	1	€0.02
R8	15K Ω	739-7143	Radionics	1	€0.01
S1, S2, S3, S4	Switch - White	B3W-1050	Radionics	4	€2.60
S5	Switch - Yellow	SKHHCRA010	Radionics	1	€0.50
S6	Switch - Brown	DTS63NV	Radionics	1	€0.09
C6	1nF	DE2E3KY102MA3BM02F	Radionics	1	€0.20
R8, R11	220 Ω	RS-Carbon-220R-5%-0.25W	Radionics	2	€0.01
R9	22k Ω	RS-Carbon-22k-5%-0.25W	Radionics	1	€0.01
R10	47k Ω	RS-Carbon-47k-5%-0.25W	Radionics	1	€0.01
Q1, Q2	BC107 Transistor	610-BC107	Mouser	2	€0.84
N/a	PCB Board	RE520-HP	Multisort Elek.	1	€1.50
				Total Cost	€30.37

Problems encountered

The major problem that was encountered throughout the process of developing this prototype was with the 3.3V supply. The pins soldered into the board were of the wrong orientation which caused the power supply to output the wrong values. By troubleshooting and checking the correct pin operation, we were able to amend the error and finally got the desired results from the supply.

MSP430 pin – the problem occurred at the digital pin (p1.6) where I tried to detect a signal through it and the pin was unresponsive to commands and testing. It turns out that the specific pin of the microcontroller was damaged due to overuse from the past testing which render the pin unusable. The issue was resolved through the help of the Technical Officer of IT Tallaght which provided me with a new chip.

MSP430 Reset Switch – The switch was initially in place in series with a 1nF capacitor connected to 47kΩ. The schematic of the MSP430 reset switch from the datasheet was overlooked that results to an unresponsive reset function. By setting the capacitor in parallel with the switch, the error was amended and resulted to a fully functional reset switch as intended.

Hardware and Operation Testing

Objective

- To test the output voltage of the power supply (5 V and 3. 3 V)
- To test and display values to the LCD display using MSP430
- Measure the voltages related to the switch
- To test the reset switch voltage behaviour associated with MSP430
- To test and observe detection of movement using function generator
- To test and verify working order of the bike computer prototype

Equipment Used

- Multi-meter, Prototype Bike Computer, LCD display, MSP430G2553, DC Power supply, Function Generator

Power supply

Procedure:

- Supply a 9 V DC power to the circuit.
- Observe and measure each output voltage from the diode, 3.3 V voltage regulator and 5 V voltage regulator. Tabulate results.

Schematic:

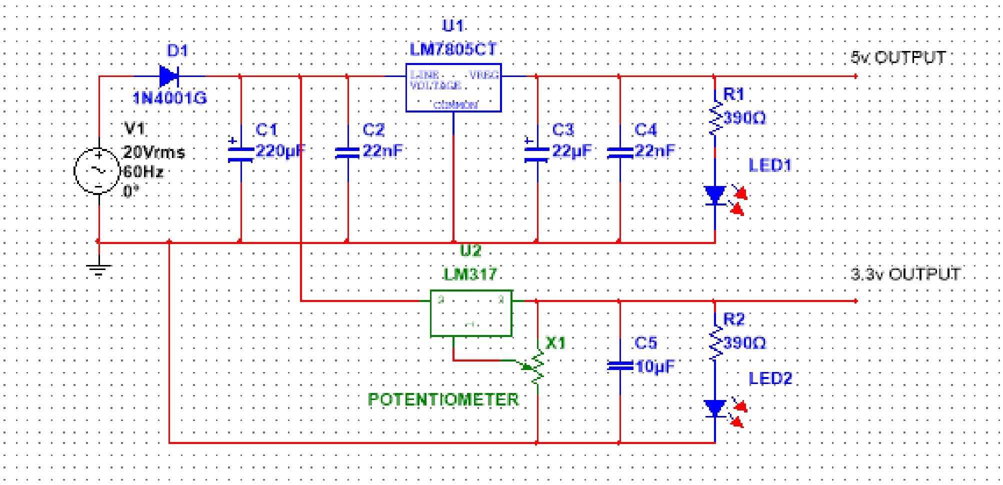


Figure 13: Power Supply (5 V and 3.3 V)

Results:

	Measured Value	Expected Value
Diode	8.34 V	8.3 V
3.3 V supply	3.3 V	3.3 V
5 V supply	4.95 V	5 V

Table 4: Power Supply Testing Results

Comments:

The measured values are within the expected result. Since the diode has a voltage drop of 0.7 V, the result of the test agrees to the measured value. 3.3 V supply are a match with the expected value indicating that the voltage regulator of 3.3 V is in working order. The 5 V supply has a slight result variation of 0.05 V from the expected voltage. This result is minimal difference may be due to the tolerance of the components used caused the voltage drop thus, this variation is neglectable. The power indicator LED's turns on as expected. Overall, all components are in working order and the power supply unit as a whole for the project is working correctly as intended.

LCD and Microcontroller

Procedure:

- Connect the correct pins of the LCD as illustrated below to the I/O pins of the MSP430. Follow the diagram illustrated below and program the MSP430 to display "Carl Sagrado".
- Adjust contrast to desired level.
- Retain LCD and MSP430 setup to be used to test the switches and sensors

Schematic:

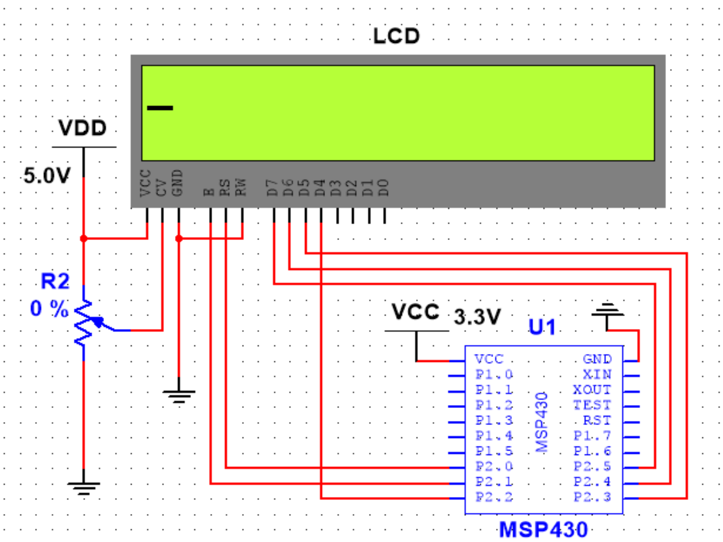


Figure 14: LCD and MSP430 Testing

To kick start the project testing with our microcontroller, we are going to instruct the LCD to display information using the MSP430. As seen in figure 14, the LCD is powered by 5 V supply and the microcontroller is powered by 3.3V supply. The contrast of the LCD is connected to a variable resistor which will take an input voltage from the same 5 V supply. We are going to use a 4-bit interface mode to save pin and therefore, we are going to connect D4 to D7 to the pins to P2.2, P2.3, P2.4, P2.5 respectively. The enable pin and the register select pin of the LCD will be connected to P2.1 and P2.0.

Results/Comments:

The testing of the LCD and MSP430 was a success. We were able to display texts on the display as intended. Once the library of the Liquid Crystal display was imported into the program, we were able to call the functions in the library that allows us to communicate and control the LCD display. Overall, the testing was a complete and we have met our objective. The pins for both LCD and MSP430 will remain reserved for the displaying text into the LCD and further functionalities will be added at the later stage which will also be displayed in the display. The code for this test is available at the appendices page.

Switches and Sensors

Procedure:

- Build the switches as it is in the schematic and supply 3.3 V into the circuit.
- Connect P1_4 of the msp430 to the switches as seen on the schematics.
- Measure voltage associated to each switch when pressed and tabulate results
- Program the MSP430 to detect which button is pressed and display on the screen.

Schematic:

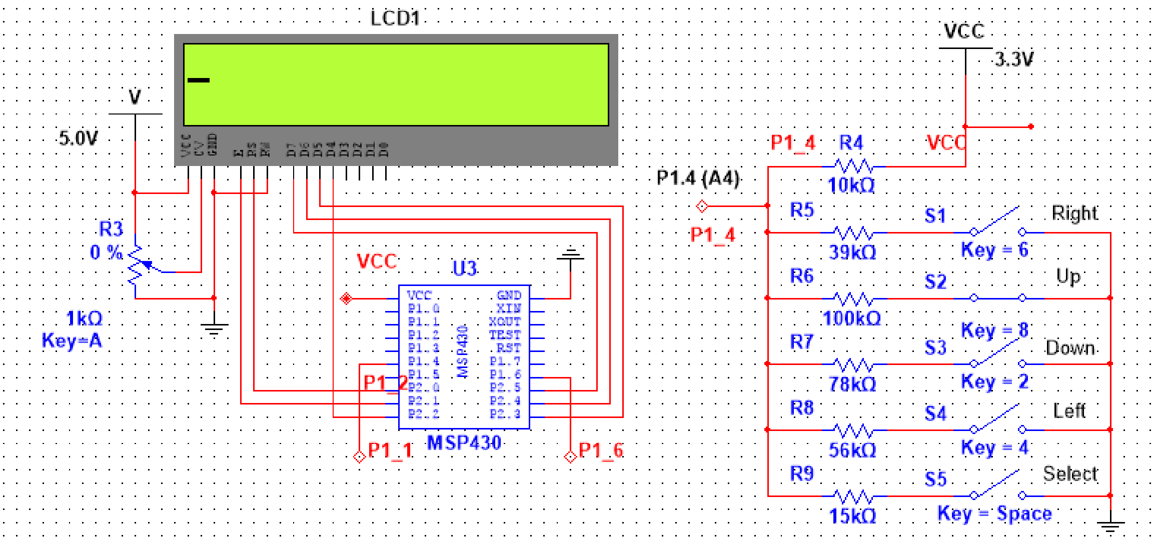


Figure 15: LCD, MSP430 and Switches

Results:

Schematic ID	Measured Voltage Value	Display to LCD
S1	2.627 V	Right
S2	3. 0 V	Up
S3	2.925 V	down
S4	2.8 V	Left
S5	1.98 V	Select

Table 5: Switch Tabulated Results

Comments:

As the table 4 shows, the measured value of the switches is within expected range of the theoretical value. It is noted that when the switches are all high, the value of the voltage input would remain 3.3 V, since source is not grounded. The variation of the voltage value during each switch press allows us to easily determine and capture these specific events using the microcontroller which are embedded with pre-defined set of instructions to react or trigger when a voltage value when a condition is met.

By reading the analog signal passed to the P1_4 of MSP430, we were able to calculate the voltage value by using the following formula:

$$V_{out} = Sample\ Value * \frac{V_{ref}}{2^n}$$

where:

Sample Value = analog signal

V_{ref} = reference voltage (3.3 V)

n = number of bits

Following the testing process and implementation of the code with the above formula into the MSP430, we were able to convert the analog signal value to a voltage value to which confirms the initial manual measurements for each pressed button. To further test the ability of the microprocessor, we set a condition at which the microcontroller listens at any event the voltage value (V_{out}) changes. Once it changes, the voltage value is compared to the conditions set to which when the value meets a condition, it will trigger a set of instruction, one of which is to display appropriate switch name into the LCD.

Overall, this test procedure was a success with no errors or problem encountered.

Movement Sensor

Procedure:

- Build the sensor as it is in the schematic and supply 3.3 V into the circuit.
- Using the MSP430, program the micro controller to perform task using interrupt at the rising edge of the wave.

Schematic:

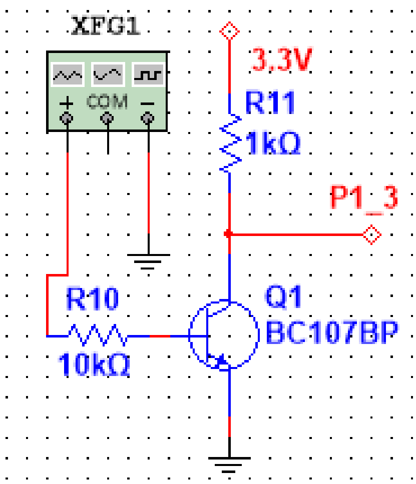


Figure 16: Movement Sensor

Results:

To further understand the how attachInterrupt function operate, we tested and visualize the result using the launchpad Green led. The test was processed by sending a square wave to the base of NPN transistor which will be digitally read by the microcontroller pin at the collector of the transistor. In doing so, a pulse signal is received by the pin and by using an attachInterrupt function, we are able to instruct the microcontroller to interrupt the normal process to execute the instruction set in blink function. The function only contains a simple Boolean inverter that inverts any assigned current value to the variable and in turn changed it to opposite value.

The visual representation as seen in figure 17 illustrates that, interrupt function serves as a timing function that reacts in relative to pulses. With this function, it ensures that the program no movement is a miss, as it reacts to real-time events.

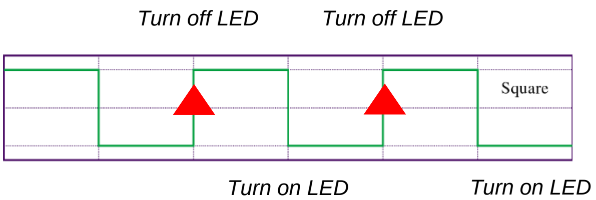


Figure 177: Visual representation of LED activity in relation to the pulse

Comments:

In relation to our Bike Computer project, using interrupt function would be an efficient way of reading the activity of the wheel. By utilising the function, it would free up activity inside the loop as the attachInterrupt operate at an event. In addition to having using an interrupt, this avoids the typical conundrum of using a delay inside the loop which can often cause a mis-calculation or missed reading to the system delay. Wheel simulation or using an actual reed switch with wheel to check the speed of the bike, interrupt function is highly applicable to be used in both scenarios.

With the success of this test, we can now use and be able to execute the interrupt function to give us our desired wheel simulation for this project. Simulating the wheel, we can set the rising edge of the wave to be a full revolution of the wheel which gives us the revolutions per minute.

Code for this test is available at the appendices of this report.

Operational Testing of the Prototype Bike Computer

Objective:

- To fully test the prototype bike computer
- To document any problems encountered during the testing
- To visually and physically test menu, features and other functionalities available on the prototype

Procedure:

- Supply 9 Volts power to the power input of the prototype
- Provide Tabulated results of the voltage output of 5 V and 3.3 V regulators
- Visually check the LCD display is powered on, adjust contrast from low to high and high to low
- Using a function generator, supply a 1 Hz square wave into the movement sensor input and observe results in the LCD.
- Test the timer by toggling each switch related to the feature
- Proceed to the toggle the menu switch and test brightness and contrast
- Adjust the brightness of the LCD by toggling the left and right switch and turn the contrast potentiometer to change the contrast level of the switch.

Results:

Power Supply

Table 6:Final Testing - Power Supply

	Measured Value (V)	Expected (V)
5 V	4.95 V	4.95 V
3.3 V	3.3 V	3.3 V

As seen in table 5, the voltage output of the power supply unit is similar to the previous test done. No errors were found during the making of this circuit and no closed circuit was found.

LCD Display

The LCD powered on as we put 9-volt power into the source. Since the LCD is connected to the 5 V supply, The LCD is expected to function as intended. The display on the LCD is what we have expected result following the debugging of the code in msp430. By toggling the adjust contrast as instructed, we were able to visually observe the result of the contrast adjustment. No error was found on this testing.

Movement Sensor

Placing one 1 Hz signal on the input, we were able to observe and expected result of the bike computer. At the rising edge of the wave, the interrupt triggers which takes the time difference during each rise of the wave and was calculated to get the linear velocity. The result is displayed on the LCD, which meets the intended purpose and thus, concluded that the sensor is in working order.

Timer Testing

The timer works by displaying elapse time in seconds. By pressing the left switch (start/pause) and compare it with a timer app in a mobile phone, we were able to verify that the timer works in seconds in real time. The same button pauses, and the right switch stops and resets the timer. As observed during testing the functionality added to the bike computer works and in sync with real time, which concludes the testing of the timer.

Brightness Adjust and Contrast Level display

To gain access to the brightness adjust section, we pressed the select/menu switch to display the second menu. The brightness adjust works by incrementing or decrementing bar display by one. The left switch (adjust -), decrements the brightness value while the right switch (adjust +) increments the value. The result of the adjustment is visible and was observed in the LCD display. It was observed that as we toggle the switches, the brightness of the display varies. This test was a success as the intended brightness adjust works perfectly which satisfies the testing objective.

By pressing the down switch, we have gained access to the Contrast level display. The intended purpose of this display is to visualize the level of the contrast as we toggle the contrast potentiometer. As observed during this testing, the level bar of the contrast increases and decreases as we toggle the variable resistor low to high and high to low. The only difficulty in this testing is that when reaching the mid contrast level, the display on the circuit would be difficult to see. However, following this test it is concluded that the contrast level display works which meets the objective of the testing.

The source code of the whole project is printed in the appendices section.

Conclusion / Recommendation

Following the multiple tests done on the system, the prototype bike computer is found to be fully functional. The voltage regulators are outputting desired voltage which results are supplies the rest of the components. The functionalities of the bike computer are displayed in the display and the features added to the device works flawlessly.

In conclusion, this project has provided me a great deal experience and confidence in in dealing with components to build an electronic device. As an electronic engineering student, this project has provided great learning curve which I believe is very beneficial to the career that I am pursuing. This project has brought in new sets of skills especially in fault finding, component datasheet sourcing, time keeping and building projects from scratch.

While the system itself is working, it could use an improvement in the display, to use a larger display could be more beneficial so that we can display more information to the user. As an upgrade, the system could also benefit from Bluetooth communication and a GPS tracking technology to allow users to save their cycle journey to their phone which they could track and analyse in the future.

Bibliography

[1] Texas Instruments, “LM317 3-Terminal Adjustable Regulator,” Texas Instruments Incorporated, 2018. [Online]. Available: http://www.ti.com/lit/ds/slvs044x/slvs044x.pdf. [Accessed 07 12 2018].

[2] “16×2 Lcd Display Module Pinout Datasheet 16 Pin Lcd,” [Online]. Available: http://jebbush.co/16-pin-lcd/16x2-lcd-display-module-pinout-datasheet-16-pin-lcd/. [Accessed 07 12 2018].

[3] “16x2 LCD Module,” 30 08 2017. [Online]. Available: https://components101.com/16x2-lcd-pinout-datasheet. [Accessed 08 12 2018].

[4] Farnell, “Alphanumeric LCD#s,” [Online]. Available: https://elearning.it-tallaght.ie/mod/resource/view.php?id=290518. [Accessed 08 12 2018].

[5] Texas Instruments, “MIXED SIGNAL MICROCONTROLLER,” Texas Instruments Incorporated, 2013. [Online]. Available: http://www.ti.com/lit/ds/symlink/msp430g2553.pdf. [Accessed 09 12 2018].

[6] National Academy of Sciences, Industrial Environmental Performance Metrics: Challenges and Opportunities, N. A. o. E. N. R. Council, Ed., Washington D.C.: The National Academies Press, 1999, p. 107.

[7] E. Williams, “Environmental Impacts in the Production of Personal Computers,” Springer International Publishing AG, 2003. [Online]. Available: https://link.springer.com/chapter/10.1007/978-94-010-0033-8_3#citeas. [Accessed 08 12 2018].

[8] Electronics TakeBack Coalition, “Where’s The Harm – From Materials Extraction?,” Electronics TakeBack, [Online]. Available: http://www.electronicstakeback.com/toxics-in-electronics/wheres-the-harm-extraction/. [Accessed 08 12 2018].

[9] E. D. Williams, R. U. Ayrs and M. Heller, “The 1.7 Kilogram Microchip: Energy and Material Use in the Production of Semiconductor Devices,” American Chemical Society, 25 08 2002. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/es025643o. [Accessed 08 12 2018].

[10] Electrolube, “Environmental and End-User Concerns: Responsible Manufacturing in the Electronics Industry,” HK Wentworth Ltd, 2017. [Online]. Available: https://www.electrolube.com/technical-articles/environmental-and-end-user-concerns-responsible-manufacturing-in-the-electronics-industry/. [Accessed 08 12 2018].

[11] Union of Concerned Scientists, “Environmental Impacts of Renewable Energy Technologies,” Union of Concerned Scientists, 2017. [Online]. Available: https://www.ucsusa.org/clean-energy/renewable-energy/environmental-impacts#.WixR9kpl9PY. [Accessed 08 12 2018].

[12] J. Cook, “Waste Heat vs Greenhouse Warming,” Sketical Science, 27 07 2010. [Online]. Available: https://skepticalscience.com/Waste-heat-vs-greenhouse-warming.html. [Accessed 08 12 2018].

[13] Energy Star, “ENERGY STAR Product Specification Development Process Description,” Energy Star, 2017. [Online]. Available: https://www.energystar.gov/partner_resources/product_specification_development_process. [Accessed 08 12 2018].

[14] Carnegie Mellon University, “How Technology can Harm the Environment,” Carnegie Mellon University, 2017. [Online]. Available: http://www.carnegiecyberacademy.com/facultyPages/environment/issues.html. [Accessed 07 12 2018].

[15] Elprocus, “Rectifier Diode Circuit Working And Its Applications,” Eletronics Projects Focus, 2013. [Online]. Available: https://www.elprocus.com/rectifier-diode-working-applications/. [Accessed 7 12 2018].

[16] niiraz, “All About IC 7805 | Voltage Regulator,” eletronicsforu.com, 14 10 2017. [Online]. Available: https://electronicsforu.com/resources/learn-electronics/7805-ic-voltage-regulator. [Accessed 07 12 2018].

[17] Aspencore, “Variable Voltage Power Supply,” Electronics Tutorials, 2018. [Online]. Available: https://www.electronics-tutorials.ws/blog/variable-voltage-power-supply.html. [Accessed 07 12 2018].

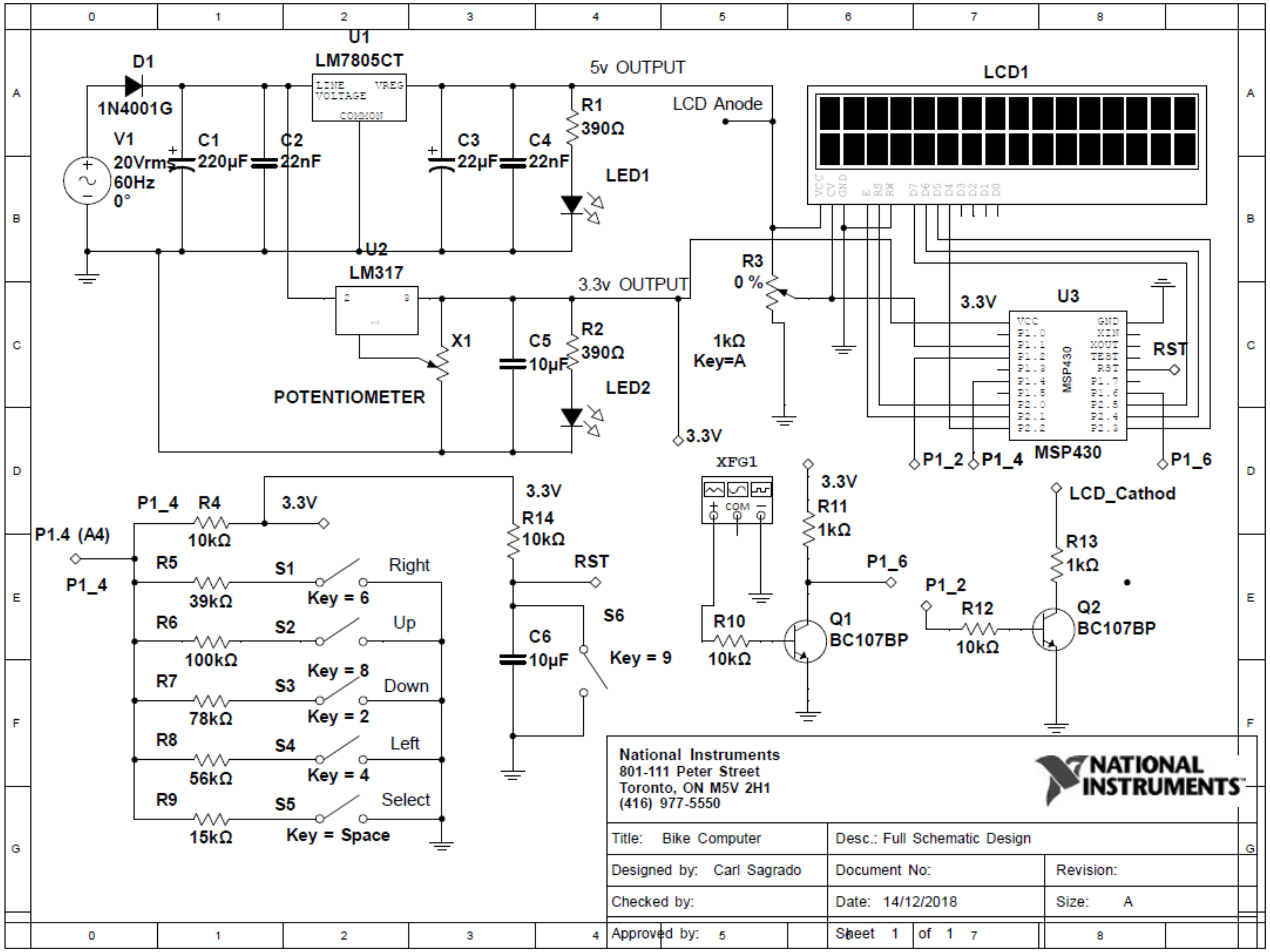
[18] EFY Group, “How 16×2 LCDs work | Build a basic 16×2 character LCD,” Electronicsforu, 4 12 2018. [Online]. Available: https://electronicsforu.com/resources/learn-electronics/16x2-lcd-pinout-diagram. [Accessed 07 12 2018].

[19] J. Kugathasan, “Digital Safe Using MSP430G2553 Microcontroller,” Sri Lanka Institute of Information Technology, 12 2017. [Online]. Available: https://www.researchgate.net/publication/322049638_Digital_Safe_Using_MSP430G2553_Microcontroller. [Accessed 08 12 2018].

[20] ranga.nr, “5 Bad Effects of Technology in Environment,” Mind Controversy, 17 02 2015. [Online]. Available: http://www.mindcontroversy.com/effects-of-technology-in-environment/. [Accessed 08 12 2018].

[21] “Environmental Guidelines for Electronics Manufacturing,” World Bank, Environment Department, 1997. [Online]. Available: https://www.miga.org/documents/ElectronicsManufacturing.pdf. [Accessed 08 12 2018].

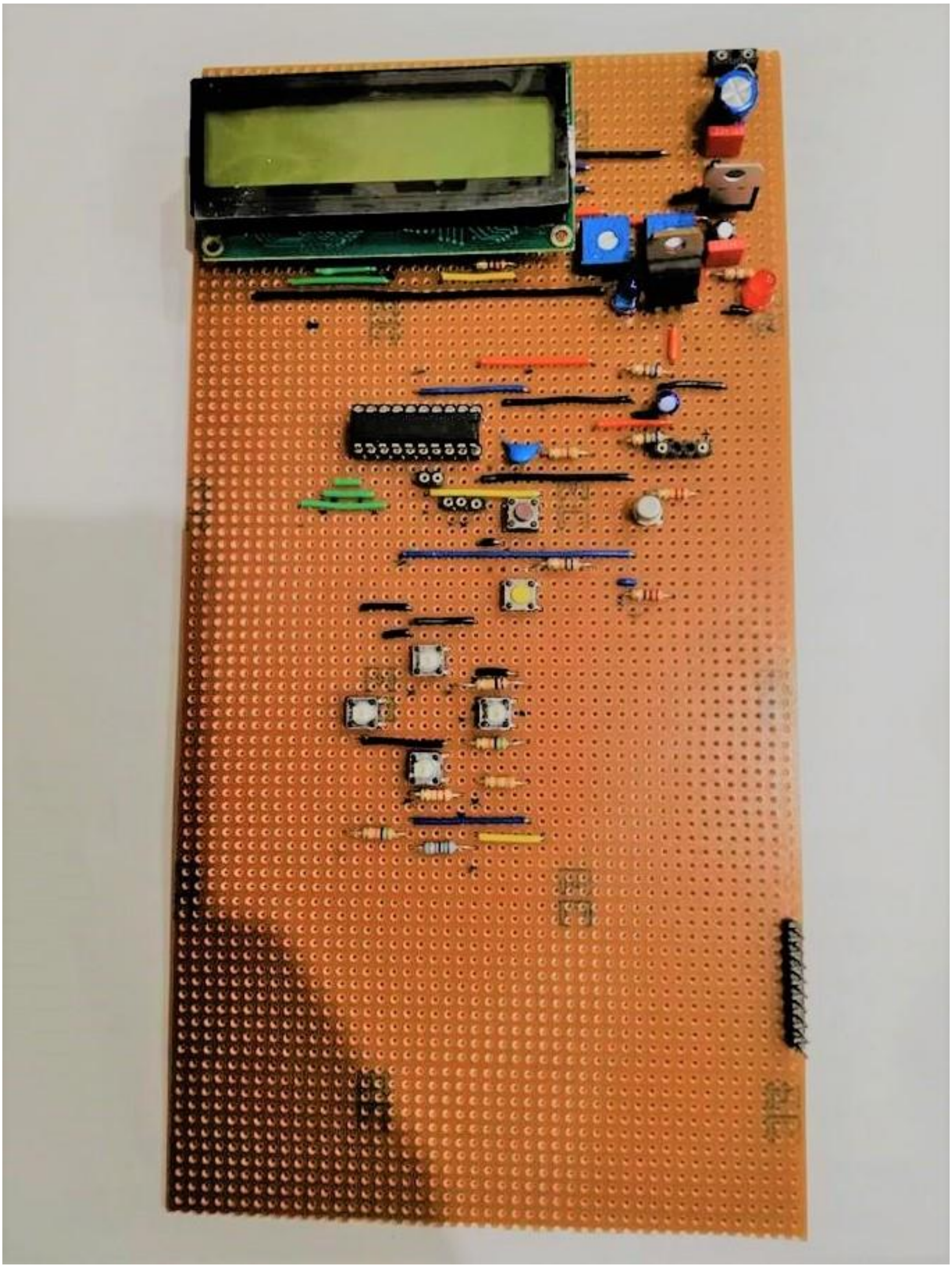
Appendices



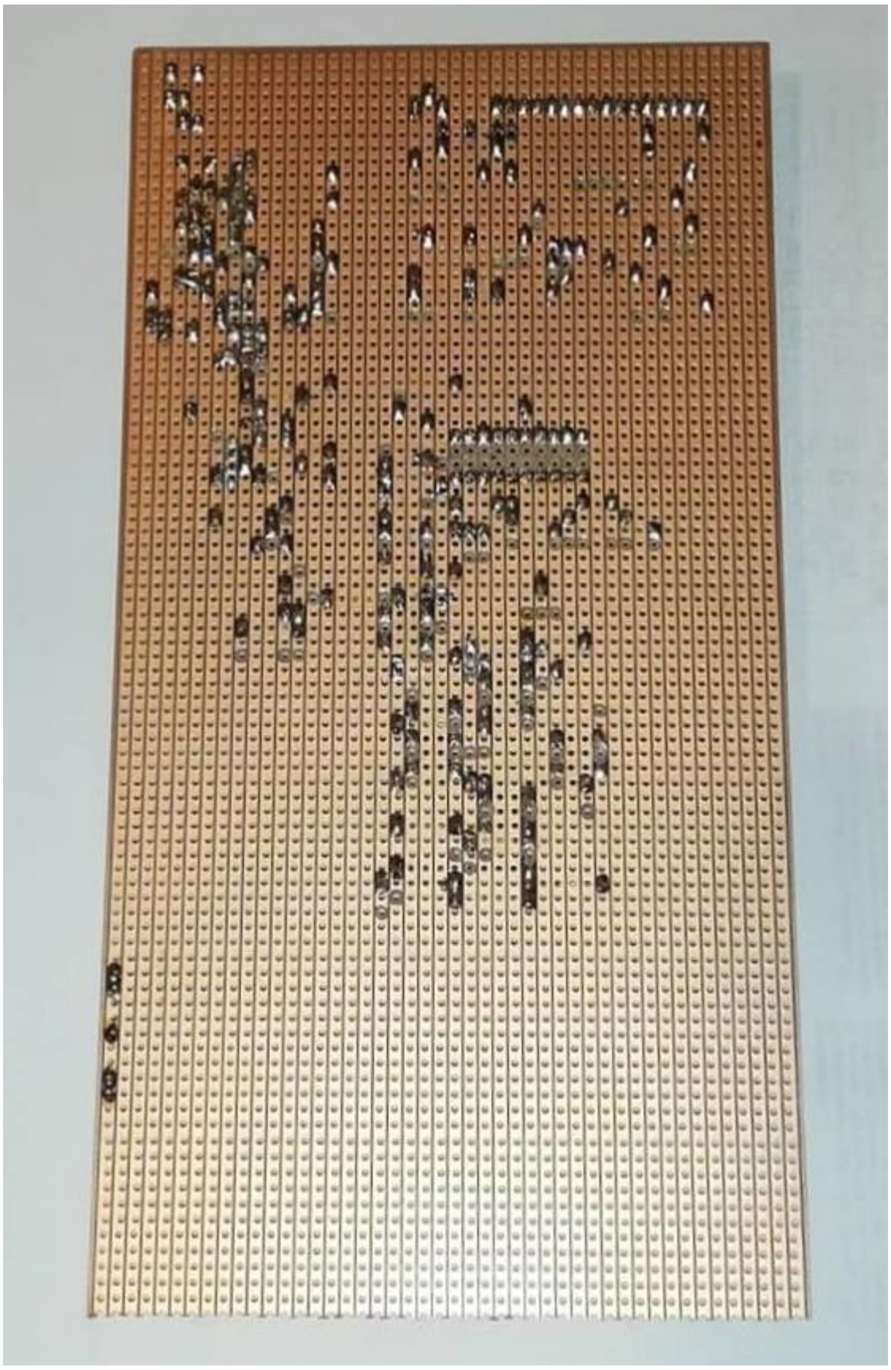
Full Schematic Diagram of Bike Computer

PCB Artwork

Top View



Bottom View



Useful information

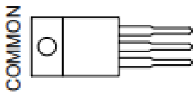
5.0 V Voltage Regulators

µA7800 SERIES
POSITIVE-VOLTAGE REGULATORS

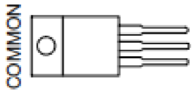
SLVS056J – MAY 1976 – REVISED MAY 2003

- **3-Terminal Regulators**
- **Output Current up to 1.5 A**
- **Internal Thermal-Overload Protection**
- **High Power-Dissipation Capability**
- **Internal Short-Circuit Current Limiting**
- **Output Transistor Safe-Area Compensation**

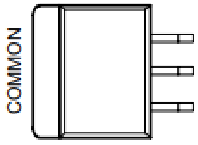
KC (TO-220) PACKAGE
(TOP VIEW)



KCS (TO-220) PACKAGE
(TOP VIEW)



KTE PACKAGE
(TOP VIEW)



description/ordering information

This series of fixed-voltage integrated-circuit voltage regulators is designed for a wide range of applications. These applications include on-card regulation for elimination of noise and distribution problems associated with single-point regulation. Each of these regulators can deliver up to 1.5 A of output current. The internal current-limiting and thermal-shutdown features of these regulators essentially make them immune to overload. In addition to use as fixed-voltage regulators, these devices can be used with external components to obtain adjustable output voltages and currents, and also can be used as the power-pass element in precision regulators.

ORDERING INFORMATION

T_J	$V_{O(NOM)}$ (V)	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 125°C	5	POWER-FLEX (KTE)	Reel of 2000	µA7805CKTER	µA7805C
		TO-220 (KC)	Tube of 50	µA7805CKC	µA7805C
		TO-220, short shoulder (KCS)	Tube of 20	µA7805CKCS	
	8	POWER-FLEX (KTE)	Reel of 2000	µA7808CKTER	µA7808C
		TO-220 (KC)	Tube of 50	µA7808CKC	µA7808C
		TO-220, short shoulder (KCS)	Tube of 20	µA7808CKCS	
	10	POWER-FLEX (KTE)	Reel of 2000	µA7810CKTER	µA7810C
		TO-220 (KC)	Tube of 50	µA7810CKC	µA7810C
	12	POWER-FLEX (KTE)	Reel of 2000	µA7812CKTER	µA7812C
		TO-220 (KC)	Tube of 50	µA7812CKC	µA7812C
		TO-220, short shoulder (KCS)	Tube of 20	µA7812CKCS	
	15	POWER-FLEX (KTE)	Reel of 2000	µA7815CKTER	µA7815C
		TO-220 (KC)	Tube of 50	µA7815CKC	µA7815C
		TO-220, short shoulder (KCS)	Tube of 20	µA7815CKCS	
	24	POWER-FLEX (KTE)	Reel of 2000	µA7824CKTER	µA7824C
		TO-220 (KC)	Tube of 50	µA7824CKC	µA7824C

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

3.3 V Voltage Regulator

Product Folder | Sample & Buy | Technical Documents | Tools & Software | Support & Community

TEXAS INSTRUMENTS

LM317

SLVS044X – SEPTEMBER 1997 – REVISED SEPTEMBER 2016

LM317 3-Terminal Adjustable Regulator

1 Features

- Output Voltage Range Adjustable From 1.25 V to 37 V
- Output Current Greater Than 1.5 A
- Internal Short-Circuit Current Limiting
- Thermal Overload Protection
- Output Safe-Area Compensation

2 Applications

- ATCA Solutions
- DLP: 3D Biometrics, Hyperspectral Imaging, Optical Networking, and Spectroscopy
- DVR and DVS
- Desktop PC
- Digital Signage and Still Camera
- ECG Electrocardiogram
- EV HEV Charger: Level 1, 2, and 3
- Electronic Shelf Label
- Energy Harvesting
- Ethernet Switch
- Femto Base Station
- Fingerprint and Iris Biometrics
- HVAC: Heating, Ventilating, and Air Conditioning
- High-Speed Data Acquisition and Generation
- Hydraulic Valve
- IP Phone: Wired and Wireless
- Intelligent Occupancy Sensing
- Motor Control: Brushed DC, Brushless DC, Low-Voltage, Permanent Magnet, and Stepper Motor
- Point-to-Point Microwave Backhaul
- Power Bank Solutions
- Power Line Communication Modem
- Power Over Ethernet (PoE)
- Power Quality Meter
- Power Substation Control
- Private Branch Exchange (PBX)
- Programmable Logic Controller
- RFID Reader
- Refrigerator
- Signal or Waveform Generator
- Software Defined Radio (SDR)
- Washing Machine: High-End and Low-End
- X-ray: Baggage Scanner, Medical, and Dental

3 Description

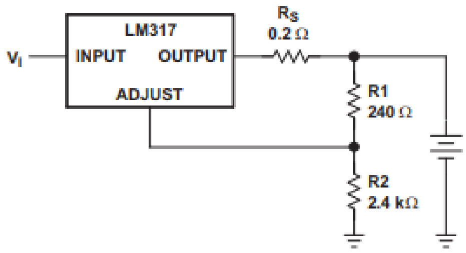
The LM317 device is an adjustable three-terminal positive-voltage regulator capable of supplying more than 1.5 A over an output-voltage range of 1.25 V to 37 V. It requires only two external resistors to set the output voltage. The device features a typical line regulation of 0.01% and typical load regulation of 0.1%. It includes current limiting, thermal overload protection, and safe operating area protection. Overload protection remains functional even if the ADJUST terminal is disconnected.

Device Information(1)

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM317DCY	SOT-223 (4)	6.50 mm × 3.50 mm
LM317KCS	TO-220 (3)	10.16 mm × 9.15 mm
LM317KCT	TO-220 (3)	10.16 mm × 8.59 mm
LM317KTT	TO-263 (3)	10.16 mm × 9.01 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Battery-Charger Circuit

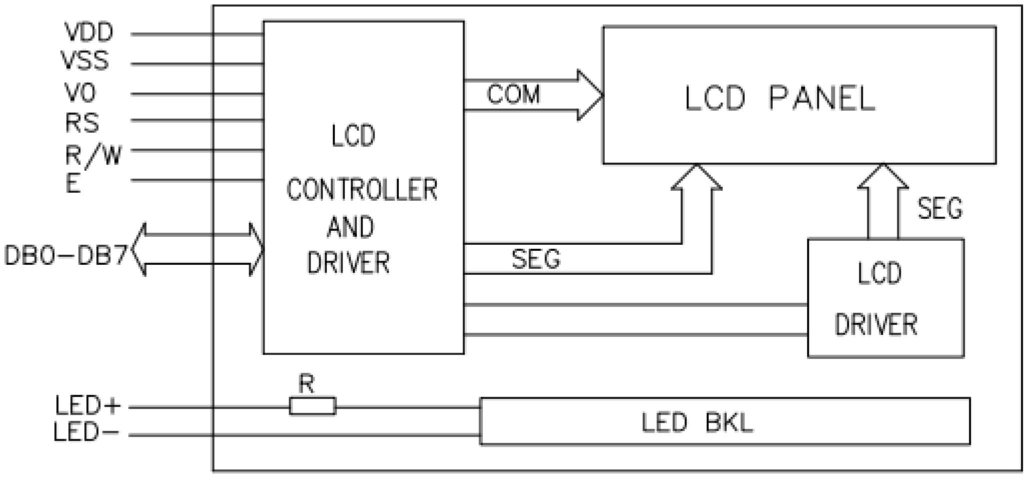


16x2 Liquid Crystal Display

4. Absolute maximum ratings

Item	Symbol	Standard			Unit
Power voltage	V_{DD}-V_{SS}	0	-	7.0	V
Input voltage	V_{IN}	VSS	-	VDD	
Operating temperature range	V_{OP}	0	-	+50	℃
Storage temperature range	V_{ST}	-10	-	+60	

5. Block diagram



6. Interface pin description

Pin no.	Symbol	External connection	Function
1	Vss	Power supply	Signal ground for LCM
2	V_{DD}		Power supply for logic for LCM
3	V_0		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	DB0~DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11~14	DB4~DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU
15	LED+	LED BKL power supply	Power supply for BKL
16	LED-		Power supply for BKL

Microcontroller (MSP430G2553)

MIXED SIGNAL MICROCONTROLLER

FEATURES

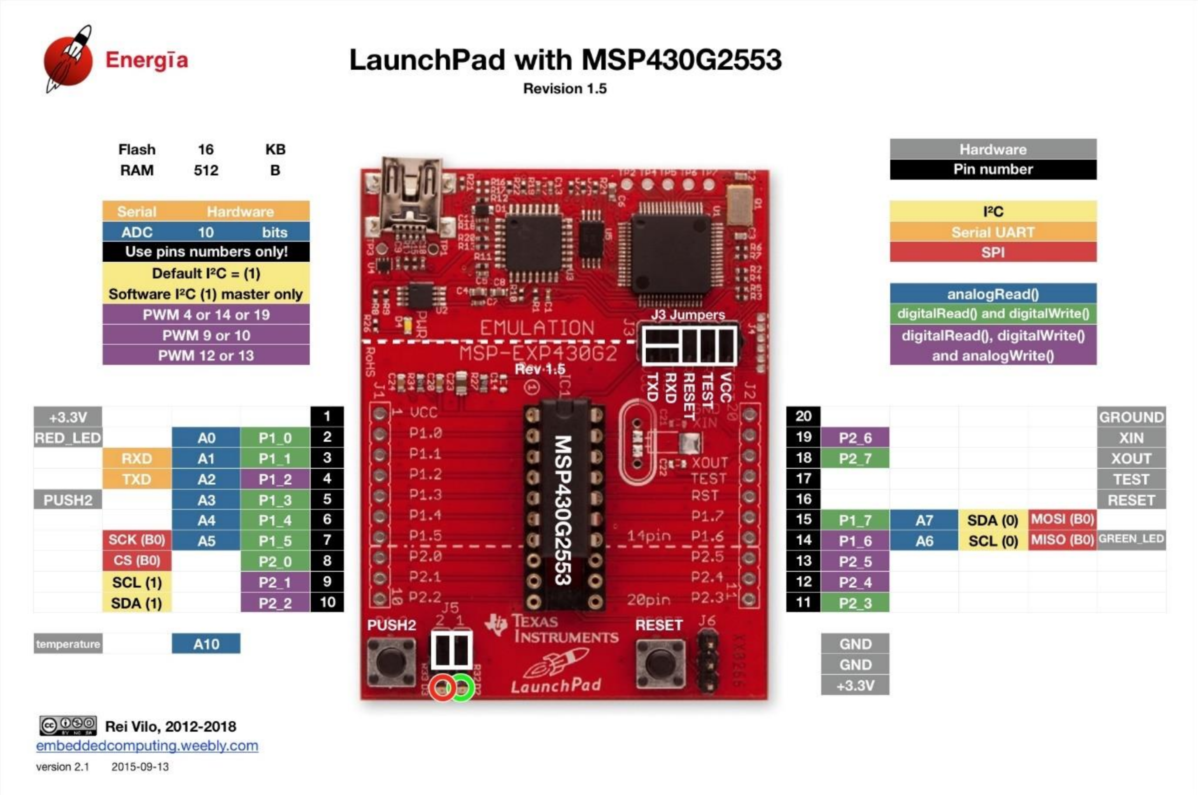
- **Low Supply-Voltage Range: 1.8 V to 3.6 V**
- **Ultra-Low Power Consumption**
  - **Active Mode: 230 µA at 1 MHz, 2.2 V**
  - **Standby Mode: 0.5 µA**
  - **Off Mode (RAM Retention): 0.1 µA**
- **Five Power-Saving Modes**
- **Ultra-Fast Wake-Up From Standby Mode in Less Than 1 µs**
- **16-Bit RISC Architecture, 62.5-ns Instruction Cycle Time**
- **Basic Clock Module Configurations**
  - **Internal Frequencies up to 16 MHz With Four Calibrated Frequency**
  - **Internal Very-Low-Power Low-Frequency (LF) Oscillator**
  - **32-kHz Crystal**
  - **External Digital Clock Source**
- **Two 16-Bit Timer_A With Three Capture/Compare Registers**
- **Up to 24 Capacitive-Touch Enabled I/O Pins**
- **Universal Serial Communication Interface (USCI)**
  - **Enhanced UART Supporting Auto Baudrate Detection (LIN)**
  - **IrDA Encoder and Decoder**
  - **Synchronous SPI**
  - **I²C™**
- **On-Chip Comparator for Analog Signal Compare Function or Slope Analog-to-Digital (A/D) Conversion**
- **10-Bit 200-ksps Analog-to-Digital (A/D) Converter With Internal Reference, Sample-and-Hold, and Autoscan (See Table 1)**
- **Brownout Detector**
- **Serial Onboard Programming, No External Programming Voltage Needed, Programmable Code Protection by Security Fuse**
- **On-Chip Emulation Logic With Spy-Bi-Wire Interface**
- **Family Members are Summarized in Table 1**
- **Package Options**
  - **TSSOP: 20 Pin, 28 Pin**
  - **PDIP: 20 Pin**
  - **QFN: 32 Pin**
- **For Complete Module Descriptions, See the *MSP430x2xx Family User's Guide* (SLAU144)**

DESCRIPTION

The Texas Instruments MSP430 family of ultra-low-power microcontrollers consists of several devices featuring different sets of peripherals targeted for various applications. The architecture, combined with five low-power modes, is optimized to achieve extended battery life in portable measurement applications. The device features a powerful 16-bit RISC CPU, 16-bit registers, and constant generators that contribute to maximum code efficiency. The digitally controlled oscillator (DCO) allows wake-up from low-power modes to active mode in less than 1 µs.

The MSP430G2x13 and MSP430G2x53 series are ultra-low-power mixed signal microcontrollers with built-in 16-bit timers, up to 24 I/O capacitive-touch enabled pins, a versatile analog comparator, and built-in communication capability using the universal serial communication interface. In addition the MSP430G2x53 family members have a 10-bit analog-to-digital (A/D) converter. For configuration details see Table 1.

Typical applications include low-cost sensor systems that capture analog signals, convert them to digital values, and then process the data for display or for transmission to a host system.


Energia
LaunchPad with MSP430G2553
Revision 1.5
Flash 16 KB
RAM 512 B
Serial
Hardware
ADC 10 bits
Use pins numbers only!
Default I²C = (1)
Software I²C (1) master only
PWM 4 or 14 or 19
PWM 9 or 10
PWM 12 or 13
Hardware
Pin number
I²C
Serial UART
SPI
analogRead()
digitalRead() and digitalWrite()
digitalRead(), digitalWrite() and analogWrite()
+3.3V 1
RED_LED A0 P1_0 2
RXD A1 P1_1 3
TXD A2 P1_2 4
PUSH2 A3 P1_3 5
A4 P1_4 6
SCK (B0) A5 P1_5 7
CS (B0) P2_0 8
SCL (1) P2_1 9
SDA (1) P2_2 10
temperature A10
20 GROUND
19 P2_6 XIN
18 P2_7 XOUT
17 TEST
16 RESET
15 P1_7 A7 SDA (0) MOSI (B0)
14 P1_6 A6 SCL (0) MISO (B0) GREEN_LED
13 P2_5
12 P2_4
11 P2_3
GND
GND
+3.3V
EMULATION
MSP-EXP430G2
Rev 1.5
J3 Jumpers
TXD RXD RESET TEST VCC
MSP430G2553
14pin
20pin
PUSH2
RESET
Texas Instruments
LaunchPad
Rei Vilo, 2012-2018
embeddedcomputing.weebly.com
version 2.1 2015-09-13

Testing Source Code

LCD and MSP430 Test Code

```
/*
  Title: LCD and MSP430 Test

  Name: Carl Sagrado
  Student no: X00084403
  Date:

  Description:
            - To test and communication link between LCD and MSP430
            - To display a text

  Version 1.0

  Hardware Required:
  * MSP430G2553
  * 10-kilohm Potentiometer
  * hook-up wire
  *Liquid Crystal Display


*/
#include <LiquidCrystal.h> //import liquid crystal library

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(P2_0, P2_1, P2_2, P2_3, P2_4, P2_5);

//P2_0 = RS, P2_1 = EN, P2_2 = 11, P2_3 = 12, P2_4 = 13, P2_5 = 14

void setup() {
  // initialize serial communication at 9600 bits per second:
  lcd.begin(16, 2); //initialize LCD screen and specify the dimensions
  lcd.cursor();  //  // turn on the cursor:
   lcd.print("Carl Sagrado"); //Display name into the LCD
}

void loop() {
// the loop routine runs over and over again forever:
}
```

Switch Test Code

```
/*
  Title: Switches Test

  Name: Carl Sagrado
  Student no: X00084403

  Version 1.0

  Description:
              - To test and display to the LCD as the switches are pressed

  Hardware Required:
  * MSP-MSP430G2553
  * 10-kilohm Potentiometer
  * hook-up wire
  *Liquid Crystal Display
  * Switches
*/
#include <LiquidCrystal.h> //import liquid crystal library

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(P2_0, P2_1, P2_2, P2_3, P2_4, P2_5); //P2_0

//P2_0 = RS, P2_1 = EN, P2_2 = 11, P2_3 = 12, P2_4 = 13, P2_5 = 14

void setup() {

  lcd.begin(16, 2); //initialize LCD screen and specifiy the dimensions
  lcd.cursor();  //  // turn on the cursor:
   lcd.print("Carl Sagrado"); //Display name into the LCD
}

void loop() {
 int sampleValue = analogRead(A4); //  read the input on analog pin A4:
 double voltage = sampleValue*0.00322265;
lcd.clear();  //clears screen and positions the cursor in the upper left
corner
if (voltage>=2.60 && voltage <=2.65){  //RIGHT
          lcd.print"Right");  //display text "right" to LCD
          delay(200);
} else if (voltage>=2.99 && voltage <=3.10){  //UP
          lcd.print("Up");  //display text "Up" to LCD
          delay(200);
} else if (voltage>=2.90 && voltage <=2.95){  //Down
          lcd.print("Down");  //display text "Down" to LCD
          delay(200);
} else if (voltage>=2.79 && voltage <=2.85){  //LEFT
          lcd.print("Left");  //display text "Left" to LCD
} else if (voltage>=1.95 && voltage <=2.00){    //Select
          lcd.print("Select");  //display text "Selecrt" to LCD
          delay(200);
}
}
```

Movement Sensor Test Code

```
/*
  Title: Initial testing of the Movement Sensor

  Name: Carl Sagrado
  Student no: X00084403


  Version 1.0

  Description:
            - To test and understand how attachInterrupt() operate
            - To visualize result of the attachInterrupt function

  Hardware Required:
  * BC107 NPN transistor
  * 1 kohm Resistor
  * 10 kohm Resistor
  * hook-up wire
  * MSP430G2553
  * Function Generator
*/


volatile int state = HIGH;  //initialize state to high
volatile int flag = HIGH;  //set condition of flag to High

void setup()
{
  pinMode(P1_6, OUTPUT);  //set the pin to an output pin
  digitalWrite(P1_6, state);  //set the P1_6 to be (high or low) depending
on the boolean value of the state variable
  attachInterrupt(P1_3, blink, RISING); // Interrupt is fired whenever the
wave rises
}
void loop()
{
  digitalWrite(P1_6, state); //LED starts ON
  if(flag) {    //check if flag is true or false
    flag = LOW;  //if flag is true, set flag to low
  }
}
void blink()
{
  state = !state;  //invert boolean value of the state
  flag = HIGH;  //set flag to high
}
```

Bike Computer - Full Source Code

```
/*
  Title: Bike Computer Source Code

  Name: Carl Sagrado
  Student no: X00084403


  Current Version 3.5

  Previous Versions

          - v1.0 - LCD display testing
          - v2.1 - Switch testing
          - v2.2 - display Switch value to LCD
          - v3.0 - Test simulated wheel
          - v3.1 - calculate time difference
          - v3.2 - display time difference to LCD
          - v4.0 - Add timer funcitionality
          - v4.1 - trigger timer using left/right switch
          - v4.2 - start/pause (LEFT Button) and stop/reset (RIGHT Button)
          - v5.0 - Add brightness control
          - v5.1 - send PWM signal to transistor for Backlight
          - v6.0 - add contrast level display
          -

  Description:
            - To apply all the codes and to fully test bike computer to
full working order in time for the DEMO


  Hardware Required:
  * MSP430G2 Launchpad
  * DC Power Supply
  * Function Generator
  * Bike Computer Prototype
*/

volatile long currentTime;  //store microprocessor time
volatile long previousTime=0;  //calculated time dependent to to attach
interrupt function
int circumference = 2;   //set circumference of the wheel to 2m

boolean right = false, left = false, up = true, down = false, btnSelect =
false;
volatile long currentSpeed; //speed
volatile long Timemillis; //time in millis
volatile long TimeSeconds; //time in seconds
volatile long  elapseMillis = 0;
volatile long  elapse = 0;//elapse time
int t = 0;

double voltage = 0;
int backlight = 5;      // variable for backlight adjustment
int brightValue = 5;    //variable used for display and for the backlight
int contrastValue = 0;  // variable to store the value coming from the
potentiometer
int intensity = 0;    //contrast intensity mapped from contrast value
```

```
#include <LiquidCrystal.h>  //caling the LCD library

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(P2_0, P2_1,P2_2,P2_3, P2_4, P2_5);

byte pBar[8] = {       //brightness and constrast custom progress bar
  B11111,
  B11111,
  B11111,
  B11111,
  B11111,
  B11111,
  B11111,
};



void setup() {

  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
  lcd.createChar(0, pBar);  //custom char
  pinMode(P1_2, OUTPUT);  //set pin 1.2 to be an output pin (backlight)
  pinMode(P1_6, INPUT_PULLUP);  //set pin to input pull up
  attachInterrupt(P1_6, wheel, RISING);  //using attach interrupt function
at the rising edge of the signal
}

void loop() {

  int sensorValue = analogRead(A4);  //Reads analog value from P1_4 (reads
condition of the SWITCHES when pressed)
  voltage = sensorValue*0.00322265;
  contrastValue = analogRead(A1);  //Reads analog value from P1_1 (From POT
to LCD that adjusts CONTRAST)
  analogWrite(P1_2,backlight); //Brightness
  // buttons();           //listen to the button clicks
  currentTime = millis();
  MenuSelection();  //calling the menu selection function
  if (btnSelect==false){  //MENU 1 *DEFAULT DISPLAY"*
    lcd.clear();  //clear the screen
    lcd.setCursor(0,0);  //default/home cursor
    lcd.print("Speed: ");  //display title
    lcd.print(currentSpeed);  //display current speed in the LCD
    lcd.print("m/s");  //add unit to speed
    lcd.setCursor(0,1);  //set the cursor to next row
    lcd.print("Elapse: ");  //display title
    lcd.print(elapse);  //display the elapse time
    lcd.print("s");  //unit in seconds
    Timer();    //Function call for the Timer 134
    delay(239); //delay to clear and display information to the LCD (This
delay is related to the timer
    //Taking into consideration the delay given to the system (timer also
takes the toll of the delay) therefore, by rationalising the delay time, it
would output desired ratio of the seconds of the timer

  }
```

```
  else {             //MENU 2 *Display adjust menu option (Brightness or
Contrast)

    if (up==true){

      Brightness();    //call the brightness function
      if (voltage>=2.60 && voltage <=2.65){ //detects when btn right is
pressed
        brightValue  = btnAdd(brightValue);   //increment brightness value
      }
      else if (voltage>=2.79 && voltage <=2.85){ //detects when btn left is
pressed
        brightValue = btnSubtract(brightValue);  //decrement brightness
value
      }
    }
    else{
      Contrast(); //call the contrast function

    }

    if (voltage>=2.91 && voltage <=2.94){  //detects when btn down is
pressed
      up=false; //set 'up' to false to change menu option (go to contrast)
    }
    else if (voltage>=2.99 && voltage <=3.10){  //detects when btn up is
pressed
      up=true;   //set 'up' to true to change menu option (go to
brightness)
    }

  }
  TimeSeconds = Timemillis*0.001;  //convert millisecond to second (1/1000)
  currentSpeed = circumference/TimeSeconds;  //calculate velocity by
distance over time

}


boolean buttonStatus (boolean x){  //invert boolean value of the button
when called

  if (x==true){  //check if value is true
    x = false;  //if true, change value to false
  }
  else {        //check if value is true
    x = true;  //if true, change value to false
  }
  //  delay(250);
  return x;  //return a boolean
}

int btnAdd (int x){  // button increment
  if (x>=10){     //if the value exceed 10 or equal to 10
    x = 10;       //by default, value is 10
  }
  else {
    x++;        //otherwise, increment the value
  }
  return x;   //return an integer value not greater than 10
```

```
}

int btnSubtract (int x){  //button decrement
  if (x>1){  //if the value is greater than 1
    x--; //proceed to subtract by 1
  }
  else {
    x = 1;  //lowest value is 1
  }
  return x;   //returnan integer value
}

int buttonCounter (int x){
  if (x>10){
    x = 0;
  }
  else {
    x++;
  }
  return x;
}

void MenuSelection(){
  if (voltage>=1.95 && voltage <=2.00){   //detects when btn select is
pressed
    btnSelect = buttonStatus(btnSelect);  //invert button select value
  }
}


void Brightness(){


  lcd.clear();   //clear the display
  lcd.setCursor(0,0); //set cursor to row 0, column 0
  lcd.print(" LCD Brightness");  //display title (move text to middle using
space
  lcd.setCursor(0,1);   //set the cursor to row 1, column 0
  backlight=map(brightValue, 0, 10,0, 255); //set the cursor row 1, column
dependent to i (either incremented or decremented)
  int pBari=map(brightValue, 0, 10, 4, 13); //take the mapped value and map
the value to be used for progress bar
  //prints the progress bar
  for (int i=3; i<pBari; i++)
  {
    lcd.setCursor(i, 1);   //set the cursor row 1, column dependent to i
(either incremented or decremented)
    lcd.write(byte(0));  //display the custom char in the display (position
dependent to the setCursor fnction
  }

  delay(200);


}

void Contrast(){

  lcd.clear();           //clear the display
  lcd.setCursor(0,0);     //set cursor to row 0, column 0
```

```
  lcd.print("  LCD Contrast");  //display title (move text to middle using
space
  lcd.setCursor(0,1);           //set the cursor to row 1, column 0
  intensity=map(contrastValue, 0, 255, 0, 10); //map the value coming from
Analog (CONTRASt)
  int smileyi=map(intensity, 0, 10, 4, 13);  //take the mapped value and
map the value to be used for progress bar

  //prints the progress bar
  for (int i=3; i<smileyi; i++)
  {
    lcd.setCursor(i, 1); //set the cursor row 1, column dependent to i
(either incremented or decremented)
    lcd.write(byte(0));  //display the custom char in the display (position
dependent to the setCursor fnction
  }
  delay(500);
}

void Timer(){

  if (voltage>=2.60 && voltage <=2.65){  //RIGHT
    right = buttonStatus(right);  //invert boolean value
    delay(100);
  }
  else if (voltage>=2.79 && voltage <=2.85){  //LEFT
    left = buttonStatus(left);  //invert boolean value
    delay(100);
  }


  elapseMillis = elapseMillis + t;
  elapse = (elapseMillis/4);

  if (left ==true){ //if left is pressed, start the time
    t=1;  //add to elapse millis each loop
    right = false;
  }
  else {
    t = 0;
  }

  if (right ==true){  //if right is pressed, timer reset
    elapse = 0; //
    elapseMillis = 0;  //
    t = 0;
    left = false;
  }
}
void wheel(){
  Timemillis = currentTime-previousTime;  //take time difference as "Time"
  previousTime = currentTime; //then assign the current time to previous
time


}
```

User Manual

The contents of this manual assume that the default settings of the bike computer are used. For your own safety, if you are not confident in dealing with electronic device or any device that involves electricity, please ask someone who is capable for guidance.

To start using the bike computer, please prepare the following:

- 9 Volt Battery
- Bike Computer Device (version ITT2018)
- Safety strip wires (included in the package)
- Wheel sensor

Quick start Guide:

Step 1: Using the safety strip wires included in the package, attach the wires to the 9-volt battery. (Notice: observe the positive and negative sign of the battery, the red wire should go into the (+) positive of the battery and black wire to the (-) negative side of the battery.)

Step 2: Next, connect the wheel sensor to the bike computer and place the other end of the sensor to the spoke of the wheel.

Step 3: Take the other part of the sensor and tape it directly at the spoke of the bike. Make sure that the two parts of the sensor is facing each other when takes the full rotation.

Step 4: Ensure that battery is placed in a safe location and switch on the Bike computer.

Step 5: You are done! The bike computer will automatically calculate your speed and will show in your display.

Using Timer

Step 1: To test the timer, simply press the left button. The bike computer will count itself in seconds which is shown in the display. To pause the timer, simply press the button again. (The left button works in two ways Start/Stop)

Step 2: To stop the timer, simply press the right button. If you press the right button again, it will reset the timer. (The right button works in two ways Stop/Reset).

Adjusting the Brightness:

To adjust the display brightness simply press the menu button once. You can increase and decrease the brightness by simply pressing the right and left button. To navigate back to the main display, press the menu button once.

Adjusting the Contrast:

To adjust the display contrast simply press the menu button once. The menu will bring you to the adjust brightness. Next, press the down button to bring you to the contrast level display, you can increase and decrease the contrast level by turning the lever in either direction. In addition to the feature, the contrast level is displayed in the LCD. To go back to brightness menu, you may press the up button otherwise, to go back to the main display, just press the menu button again.

Environmental Issues Concerning Electronic Assembly

“The Electronics industry has far transformed the facet of society. In addition to providing the basis for the information revolution, electronics enable the society’s vital support systems, including those that provide for such necessities as food, water, energy, transportation, health care, telecommunications, trade and finance.” [6]

The emerging challenges of the Electronics Industry is its impact towards the environment. And thus, the task to tackle this impact has become one of the challenges of the industry since the Electronics sector is by no means slowing down and stopping anytime.

Production

“The environmental impacts associated with a product include those that occur in the production processes to make it. These impacts occur not only at the final stage of assembling parts, but also in the production of parts and their constituent materials.” [7]

Consuming non-renewable resources including precious metals like gold are used to make electronic component e.g. Processor. The mining of such resources pollutes the water of surrounding the communities through cyanide contaminated waste ore and other abandoned mine waste including toxic metals and acid are often get released in to lakes, streams, and the ocean, killing fish and contaminating drinking water. [8]

Another concern in production of electronics such as microchips, printed circuit boards and computers is the use of many nasty chemicals during production. The industry has become globally competitive in the face of a rapidly changing technology which greatly increased the demand to produce more of the said products. The production requires a meticulous and tedious process which requires workers to be exposed with the chemicals and acids such as phosphoric acid, hydrofluoric acid, nitric acid, sulfuric ammonia, hydrogen peroxide, isopropyl alcohol, tetramethylammonium hydroxide, acetone, hexamethyldisilane, etc. which are harmful substances [9] and when processed, releases potential air emissions which include: toxic, reactive and hazardous gases that contributes to the rising global warming.

The products used to manufacture and produce technology often contain solvents called Volatile Organic Compounds, or VOC’s which are Carbon based compounds that vaporise easily at certain temperature. VOC’s affects the environment by contributing towards the formation of ground-level ozone, the main component of smog (air pollutant) to which does not only have many detrimental effects in the environment, causes health problems to anyone when over exposed. [10]

Due to the alarming affect to global warming, regulations such as REACH (Registration, Evaluation, and Authorisation of Chemicals) were set out and established to have an immediate implementation globally and companies are required to meet the standards.

Usage

Power consumption and demand has risen due to technology. Though for some, technology has aided humanity to become efficient to their work and even for some allowed them to connect to individuals from far distances. As technology helped humanity get through our primitive beings and allow us to multi-task, the end game of this advancement is the toll to our environment. For example, we always need electric energy to keep our gadgets running, to keep our lights on, etc.

Electric energy is generated by use of fossil or nuclear fuels on large scale. New renewable technologies may have been available, however, that too creates and contributes to global warming. The exact type and intensity of environmental impacts may have varies depending on the specific technology used, location, and other factors, however, it still in fact contributes to global warming [11].

Another factor arising the use of the electronic assembly is the heat generated by electronic devices. As the heat dissipates to the environment it creates carbon dioxide (CO_2), a gas that stores heat which contributes to greenhouse warming [12].

To address the issue, a program called ENERGY STAR was developed, a product specification process that relies on rigorous market, engineering and pollution savings analysis to tackle the concurring greenhouse problem. The program test, evaluates, and validates products to control and reduce greenhouse emissions. The specifications laid out by ENERY STAR aims to direct consumers to a more energy efficient product [13] which would lessen the greenhouse gas contribution to greenhouse warming.

Disposal

Electronic assembly disposal has increasingly acute as more devices invented every day rendering previous devices obsolete. These electronic wastes are known as “Techno trash” where any broken or unwanted electrical or electronic device is currently the most rapidly growing type of waste. Most of these wastes contains non-biodegradable materials, heavy metals and toxic materials that ends up in a landfill that contributes to the growing problem of land pollution, in addition to land pollution, overtime, these toxic wastes end up contaminating the water that animals, humans, plants consume. [14]

Conclusion

While electronic devices have positively impacted the society in a massive way. We must also take the account of the negative impact it does to our environment. The government and the environmental organisation all over the world have participated acted towards preserving the environment as much as we can and therefore, everyone is incumbent to participate. There are number of ways we can help the environment, such as; avoid producing any electronic assembly that has too little impact or no significance to the society, monitor waste production, producing products that are energy sufficient, use devices when necessary and turn off when not needed, separate techno trash from ordinary household trash and dispose or donate unwanted electronic devices to be properly recycled.