Cyclometer

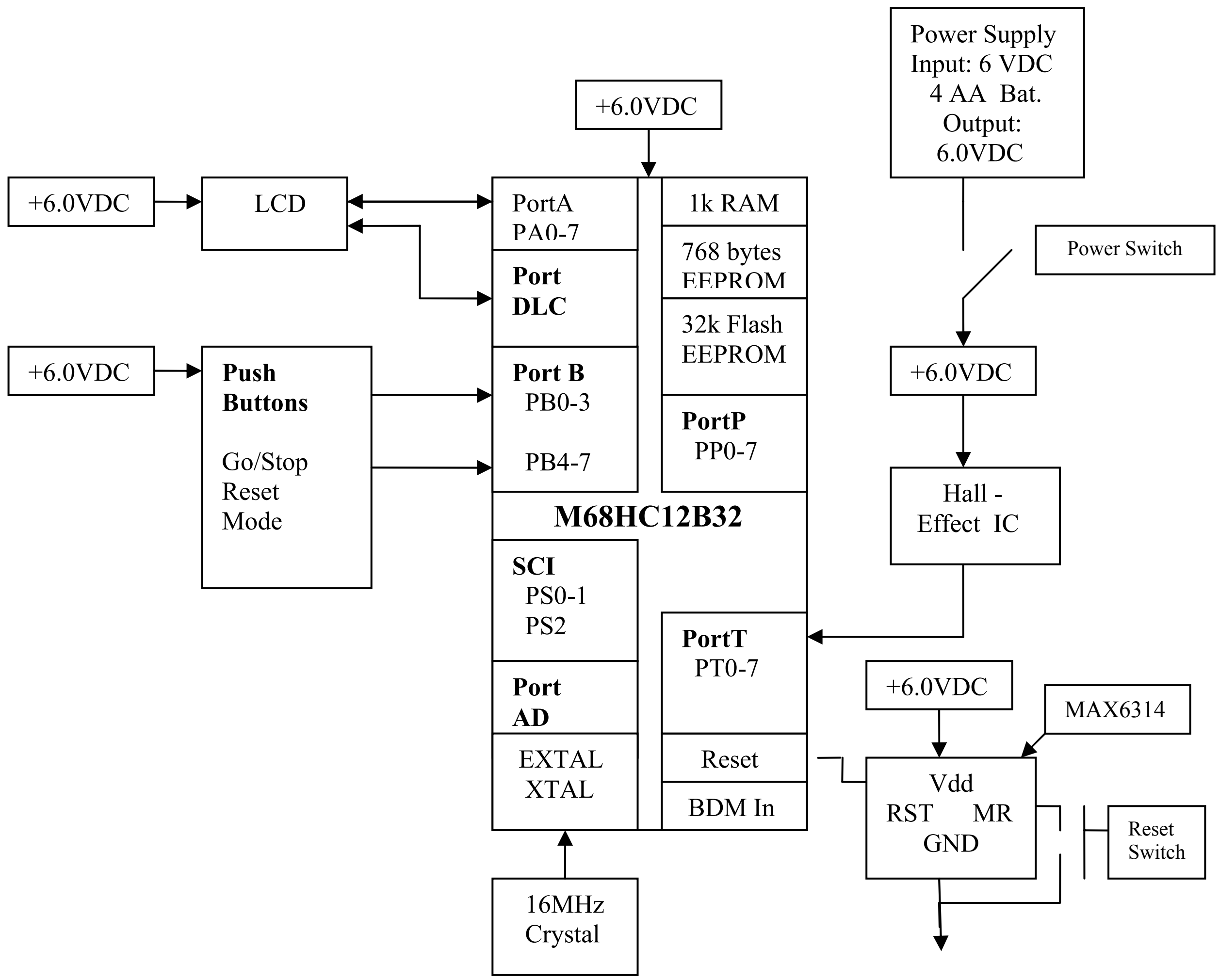
General Description:

My cyclometer will be a low power, battery operated device. Riders will be able to read their total ride time, current speed, and overall distance traveled from an LCD display. I noticed many cyclometers on the market used very small LCD displays which seemed to me like they would be difficult to read, especially while glancing down at high speeds. My cyclometer will incorporate a larger LCD display for easier viewing. Also I will be using more practical AA batteries opposed to small watch batteries.

Functional Description of Hardware:

A detailed Cyclometer hardware block diagram is shown in Figure 1.2. The primary hardware parts include the Hall-effect sensor IC, M68HC912B32 micro-controller, Liquid Crystal Display (LCD), 4-Pushbutton Interface, and a Portable Power Supply. This project is software intensive, thus the hardware list is short, and less complex. Figure 1.2 shows a diagram of the Functional Hardware design.

Figure 1.2 Functional Hardware Design



Power Supply:

For my cyclometer I'm using four AA batteries to supply a 6 volt source voltage to all my components. This seemed like the easiest method, and since the 6 volts falls within the range for my sensor, and is adequate for the HC12 and LCD, no voltage regulators are required. This is good, because voltage regulators would drain my batteries faster. I have to keep a close eye on the voltage and make sure it does not exceed 7 volts, which is above the HC12 operating voltage range which could cause problems with my Processor.

Hall-effect IC:

The Hall-effect magnetic sensor is really the key component to my cyclometer. The Hall-effect chip is a semiconductor integrated circuit utilizing the Hall-effect. It is designed to operate in the unidirectional magnetic field, especially at low supply voltage. It operates under a supply voltage range of 3.6 volts to 16 volts. Hall effect IC's are usually used as speed sensors, position sensors, rotation sensors, or small switches. In this case it will be used to detect speed and distance. The Hall-effect will keep track of each completed revolution of my mountain bike's wheel. Each time the small magnet passes the fixed sensor a small pulse will be sent to PORT T7 of the HC12 microcontroller which will count the received pulses using the pulse accumulator. There is a 1K Pull-up resistor on the out put of my Hall effect IC open-collector output pin, before it is routed to PT7. The resulting count will be used in algorithms to find speed and total distance traveled. The Hall-effect sensor is perfect for this application because it has a great operating temperate range for outdoors (-40 degrees C to +85 degrees C). It is a non-contact switch needed for the moving part of the wheel, and it will function well during the vibration of a bike ride.

M68HC912B32:

The microcontroller in my cyclometer will be performing the task of handling user interface between the push-buttons and the liquid crystal display screen. Also it will be receiving signals from the hall-effect sensor and using written code with conversion algorithms to obtain speed, distance, and time to be displayed to the user. There are smaller microcontrollers on the market and faster, more complex microcontrollers if you have the money, but the HC12 has all the features I need for this project at a decent price. The HC12 has somewhat low power consumption at 45mA, which 4AA batteries can easily power, and my familiarity with the HC12 will speed the development of my code and quicken the pace of hardware connections. The 32k bytes of Flash memory allotted for the HC12 should be large enough for my code, and the 1k byte of Ram will be used for storing temporary variables. Ports 4,5,nad 6 of the PDLC and Port A0-7 will interface with the LCD, and Port B0-3 will interface with the push buttons. Also connected to the LCD chip is a pair of fixed resistors to replace the potentiometer which supplies a constant voltage to control the display brightness of the LCD. Port T7 will handle the input capture of signals received from the Hall-effect sensor.

The BDM, Background Debug Mode Connector is connected for post-production maintenance. The BDM is connected to Gnd, and +6VDC. Also MODA, MODB, ECLK, are connected to pins PE4, PE5, and PE6. The VFP is connected to the VFP pin on the HC12, also D1 allows +6VDC to the VFP when the +12VDC is not present. The Reset is hooked up to my MAX6314 Chip for a small current draining, efficient Reset circuit. My MAX614 chip is hooked up to Gnd, Vcc, and the Reset pin on the HC12. Port B0-2 are hooked up to 3 different user interface push-buttons, each with a 10K pull-up resistor tied high. These push buttons will switch modes on the LCD, reset the trip clock, distance and speed, and also start and stop the cyclometer. Listed on Page six labled Figure 1.1 is the Memory mapping for my HC12, and the complete hardware parts list is shown on the following page.

Parts List:

Item	Qty	Description	Designators
1	2	Cap-Mono, X7R, RAD, 10%, 24pF, 50V	C1, C2
3	4	Cap-Mono, X7R, RAD, 10%, 0.1uF, 50V	C3 - C6
7	1	Crystal, 16MHz, 0.005%, HC-49/u, 18pF	X1
8	1	Diode, BZX84C5V1S-7	D1
9	4	4 AA Batteries	NA
13	1	Battery Holder	B1
14	1	Microcontroller, MC68HC912B32	U1
15	1	Reset Chip, MAX6314US4S101- T -ND	U2
16	1	LCD 16x2, MDL-16265LV	LCD1
17	1	Hall effect Sensor, DN6852A – ND	S1
18	1	Resistor, Carbon, 1MΩ, 50V, 1/8W, 1%	R1
19	1	Resistor, Carbon, 18KΩ, 50V, 1/8W, 1%	R2
20	1	Resistor, Carbon, 680Ω, 50V, 1/8W, 1%	R3
21	3	Resistor, Carbon, 10KΩ, 50V, 1/8W, 1%	R4- R6
24	1	Resistor, Carbon, 25Ω, 50V, 1/8W, 1%	R7
25	1	Resistor, Carbon, 1KΩ, 50V, 1/8W, 1%	R8
26	4	Push-Button switches, JF15SP1H	SW1 - SW4
30	1	Female Header 3 pin, connector	P1
31	1	Male Header 3 pin, connector	J1
32	1	Female Header 16 pin, connector	P2
33	1	Male Header 16 pin, connector	J2
34	1	Female Header 2 pin, connector	P3

35	1	Male Header 2 pin, connector	J3
36	1	Female Header 1 pin, Connector	P1
37	1	Male Header 1pin, connector	J1

Memory Map:

Figure 1.1

\$0000 **\$01FF**	**CPU Registers**
\$0200 **\$0700**	**Unused**
\$0800 **\$0BFF**	**On-Chip RAM**
\$0C00 **\$0CFF**	**Unused**
\$0D00 **\$0FFF**	**Byte Erasable EEPROM**
\$1000 **\$7FFF**	**Unused**
\$8000 **\$FFFF**	**Flash EEPROM** **Final Product Program**