Boston University
Electrical & Computer Engineering
EC463 Senior Design Project

First Semester Report

Re.Cycle: The Bike Computer System

Submitted to

Nima Haghighi-Mood
nhmood@bu.edu

by

Team 14
Bikeputer

Team Members

Brian Kane bkane521@gmail.com
Katherine Murphy kmurphy2@bu.edu
Ben Paolillo bppao@bu.edu
Jiwon Song jiwonsong1119@gmail.com
Todd Sukolsky tsukolsky@gmail.com

Submitted: December 11, 2012

Table of Contents

Executive Summary

ReCycle: The Bike Computer System
Team 14 – Bikeputer

Dating back to the early 19th century, the bicycle was the first major stepping stone with regard to human transportation. Some would argue that the modern-day bicycle has come a long way. However, the classic bicycle may, in fact, not be entirely “modern”. The bicycle that we know today has no diagnostics or on-board computer, leaving the rider left out of the 21st century.

Here, we introduce ReCycle: the Bike Computer System. The system integrates a touch screen display for viewing maps, bicycle diagnostics and statistics, as well as user data such as heart rate monitoring. Rather than outfitting the bicycle with twelve different components and sensors, along with messy, tangle-prone wires, ReCycle controls all of the equipment in a safe, clean and unobtrusive way. The requirements to be met include all of the above functions, as well as the ability to dump the user's statistics onto a flash drive, have a clean footprint (through the use of green mechanical power generation), and a detachable mounting device.

1.0 Introduction

Given the recent rise in the cost of fuel as well as a push towards environmentally friendly transportation methods, the bicycle has had a recent rebirth in the United States. Bike usage in Boston has increased 122% from 2007 to 2009.

Due to this fact, we believe that the bicycle is now starting to become a legitimate form of transportation. We feel that this system will be beneficial to any serious bicycle commuter, and we would like to give bicycle users the technological benefits and features that are found in any modern car.

We feel that it is imperative to seek alternate means of transportation; those that cut the costs of fuel consumption, thereby reducing pollution in major cities; those that promote the need for green, sustainable energy production; and finally, those that are ultimately safer for commuters on the roads today.

With this in mind, we are designing ReCycle to be an all-in-one solution for cyclists and bicycle-enthusiasts alike. We plan to integrate several different components with varying features. These include GPS with maps, heart rate monitoring, bicycle statistics (both odometer/speedometer), and a trips feature which tracks particular ride times with the ability to dump this data onto a flash drive.

In addition, and more importantly, we are designing ReCycle to be rechargeable. This encompasses the green aspect of our product. ReCycle will be wall-chargeable in under twelve hours, and all of the main functions will be powered for at least one hour, supplemented by at least 50% through pedaling. We are designing ReCycle to use clean, renewable energy through the use of mechanical power generation (e.g., a generator).

2.0 Concept Development

The most difficult part of this project is not individual module progression, but large scale integration of over five completely independent components. The main concern is the main processing unit.

To support the communication between so many different modules there must be a large availability of communication avenues, i.e. SPI, UART, Two-Wire, so that communication can take place near simultaneously ensuring no data will be lost. The Beagle Bone solves this, however, with it’s large number of GPIO’s and interface connections (See “System Description”).

To ensure the module is not “obtrusive” to the biker the main module, containing the screen, CPU and peripheral boards, needs to be in a casing of maximum size 4.5”x4.5”x2” mountable to the front handlebars. This size ensures the weight of the device will not add major instability to the bike and not distract the user from the road ahead. Also, to ensure the user can protect the system in case of inclement weather or safety there will be a custom connection port for all wire leading into the main module (battery voltage, sensors) so that the entire system can be taken apart in less than two minutes. Most stand-alone bike components take several minutes by themselves, so integrating so many into one module should not be any more difficult for the cyclist. The sensors, for instance, need to be zip-tied to the front fork. Because of this pesky attachment there cannot be any more mess along the connections to slow detachment down.

The module needs to contain some method of location tracking that is low power and very accurate. The accuracy of the GPS should be better than three meters. Three meters, or about nine feet, is less than the width of a normal street, so a maximum error less than the width of a street is a suitable goal. To ensure no large power consumption the GPS should not draw more than .2W of power. Most GPS modules run at 3.3V indicating a maximum current draw of ~60mA. Also, to help with the unobtrusive layout the GPS should have an on-board antenna, not a detachable one, so that it can sit in the main module and not have any outgoing wires.

Most cyclists travel within the speed range of 9mph to 22mph; the lower end represents the beach cruiser riding down the boardwalk while 22mph represents an avid cyclist on a competition bike travelling long distances. A defined accuracy of speed is to be within .25mph. This represents a 2.7% error to 1% error given the range of normal bike speeds. For the leisurely riders, speed is not a huge focal point so this error is permissible, and for those travelling upwards of 22mph a 1% error is very strong.

A resting heart rate for humans is 35bpm to 90bpm depending on their level of fitness. The easiest way to find this resting heart rate is to count how many times the heart beats for fifteen seconds, then multiply the number by four. This method yields an accurate pulse to within four beats per minute. This is our adopted standard for measuring heart rate. Some interference will occur due to vibrations on the bicycle. These vibrations could cause imprecise readings in the pulse circuit described in the next section. Because of this, four beats per minute is a solid requirement that will test the limits of the circuitry. Alternative options for was a belt the cyclist would put on around their chest that contained an RF transmitter. This RF transmitter would send the pulse to a receiver on the main PCB which would then be processed. The implementation of that system is much more complicated than the current procedure, more costly and requires more from the cyclist (strapping and un-strapping a belt).

A huge concern in this endeavor is safety. Cyclists have a reputation of being reckless on the roads and our system should not add to that reputation by distracting

the rider from the road ahead of them. To combat this safety issue the screen, when travelling over 4mph, will default to a clock screen that is locked to the user until the speed comes down below 4mph. While this inhibits the module's use, there should be no major threat to safety when using the system.

The system is logging many different things: heart rate, speed, distance and time travelled. To preserve these statistics, a "Trips" feature will be implemented to store the data in system memory on the Beagle Bone, up to 20 hours worth. To export the statistics, a USB should be plugged into the "bone" and then a file containing the information should be uploaded to it. To avoid overflow of data on the system, after 20 hours of data has been logged the user should be prompted as to which Trip(s) should be erased. Also, upon uploading of the Trip file the user should be prompted to save the information just transferred or to delete. This will allow the "bone" to operate with more free space and be more efficient.

A resistive-touch screen can come in many different forms: LED, LCD, OLED, etc. LED screens tend to cost a significant more amount than LCD screens, and the same is true for OLED screens. This leads us to search for a sunlight readable LCD screen with a refresh rate of 60Hz and 18-bit color, very standard requirements. The sun-light readable portion of this requirement is achievable via a normal LCD screen at full backlight with a custom film attached on top that increases readability. This is a low-cost solution because $40-$60 dollar LCD screens quickly become hundreds of dollars when looking for specific sunlight readable screens.

Supplementing the power of the system through the environment is effective in one of two ways for the bicycle: solar or pedal power . Solar power presents a major problem in space allocation. Even a small solar panel would have to be appropriately mounted onto the back of a bicycle and directly face the sun to be useful whatsoever. If the rider turns into the sun and the solar panel is on the rear of the bicycle, the system is no longer benefitting from the panel. This is why our system implements pedal power. Completely lossless wheels can produce 40-150W depending on the rotations per minute, so there is plenty of energy to be found through wheel rotations. Our system will use this, and in so doing, provide at least half of the power necessary to drive the system. It will charge a 12V Lead-Acid battery and power the system when enough energy is being produced. Our current circuit is rated for 500mA @ 12V, roughly 6 Watts.

Providing power to all the modules requires several regulators. It is important to drive components with non-noisy power lines that are capable of delivering enough current at their specified output voltage, i.e. the regulators cannot be overloaded. To ensure this doesn't happen a battery management system has been put in place with several five volt and three-three volt regulators. Each main component on the board will have it's own regulator, stemming from the 12V DC input of the battery and pedal power circuitry. To protect from over-drawing the battery voltage there will also be an enable

switch on the main power drawing regulators supplied via a microcontroller. This will ensure there is no dangerous happenings in the event of low power.

3.0 System Description

Within the main module will be three custom PCB's as well as the main processing unit, Texas Instruments' "Beagle Bone". The "bone" contains an ARM Cortex-A8 processor, the same processor used in many mobile phones, along with different pin out configurations. Several great features of the "bone" include it's low power compared to the Rasberry Pi (170mA vs 500mA) and it's powering capabilities. Instead of using a male wall wart plugin like the Ras Pi, the "bone" can be powered via the 5V inputs on it's main headers. Also, there is an ethernet and USB port which is idea for our requirement of USB dumpable data. The total number of GPIO pins is 96 which includes five UART connections, three SPI and three Two-Wire interfaces. The main block diagram is as follows, where the Beagle Bone is the main processing unit that all data ends up flowing through at one point.

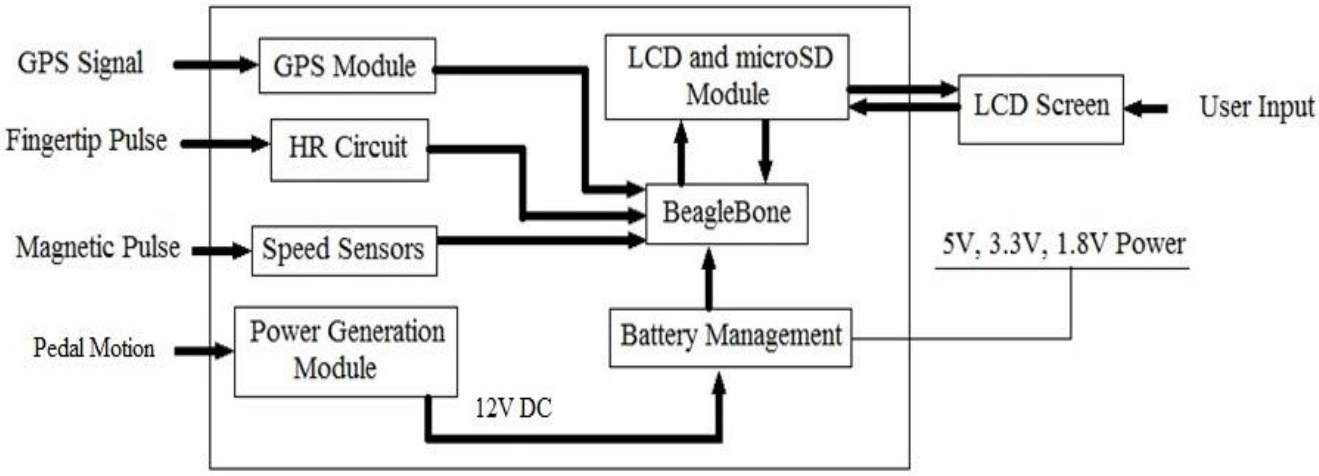


Figure 1. Block Diagram of Bike System

There will be two peripheral boards that attach to the main PCB, which then attaches to the "bone". The first of these boards contains the GPS module. The GPS is the Linx GPS which is a low power tracking solution. The GPS takes in 3.3V as well as a 1.8V backup battery input that is used to preserve the state of the non-volatile memory in the module so that on startup the GPS knows it's last location and doesn't have to go through the, at maximum, fifteen minute location lock protocol. The module will output UART data in the form of NMEA strings which contain almanac, location and satellite data. These NMEA streams will be parsed in a python script on the bone which

will pull the the location data and forward it to the LCD GUI layout for updating on the next cycle. The peak power usage of this external peripheral board is .162W with a normal operating wattage of .1W. The unique feature of this GPS is it's need to be pointing towards the sky. In other words, the chip cannot lay parallel to the ground, but needs to be perpendicular in order to keep full signal strength. If on a parallel plane it loses ~3db which correlates to half the signal strength.

The second peripheral board will contain two modules: speed-sensing circuit and the heart-rate circuit. The two circuits share a Two-Wire serial bus for communication directly to the "bone". The speed-sensing module is comprised of a magnetic sensor mounted to the front fork of the bicycle with wires travelling to a dedicated uC and a small magnet located on the spoke of the wheel. As the magnet on the spoke passes the sensor an internal gate in the sensor is closed and current travels through the attaching wires signalling an interrupt or analog input change in the microcontroller. On that interrupt or change the microcontroller will calculate the time since last interrupted and then calculate the current speed of the bicycle.

Using the speed the microcontroller can calculate total distance based on it's internal clock. The devices used to implement this is an ATtiny84A microcontroller, which consumes only 378uW with a 1.8V input at 8MHz, and the two magnets. The output of the ATtiny84A will be serial data along a shared Two-Wire bus with the heart rate microcontroller.

The heart rate system will also have the ATtiny84A to accommodate the "pulse sensor". The pulse sensor directly connects to the microcontroller providing a pulse waveform. The actual sensor uses an LED/photodiode to find pulsations and then filters this to provide a clean output. To activate the circuit the user needs to place their finger on a small, dime sized panel for over five seconds to obtain an accurate measurement The total power draw of this part of the second peripheral board is around 500uW, more than the speed-sensor circuit because of the internal power required by the sensor. As stated previously, this circuit will communicate along the same Two-Wire bus as the speed-sensor circuit flowing directly into the "bone".

The pedal power generator is a Dynamo, which produces direct current with the use of a commutator. The stator provides a constant magnetic field, and the armature consists of coils rotating within that field. The commutator reverses the connection of the windings when the potential reverses so that the current produced is DC. The total energy produced by pedal power ranges between 40-150 Watts, but our generator, the "Dynamo" outputs 6W at 12V, a typical 2A output. The generator feeds into a 12V sealed Lead-Acid battery which will be mounted below the main stem of the bicycle. This will ensure no wires touch the cyclist while biking and there will be no coiling of wires leading to a frustrating detachment process.

The main PCB will contain the battery management module, the headers for the "bone", LCD, peripheral boards, and microSD interface for .bmp files utilized by the LCD

and corresponding microcontroller. The board will bring in voltage from the battery into the battery management module and then supply the rest of the board with power. This will be a four layer board that is mounted on top of the “bone”. The main LCD header needs to be on one side of the device, comprised of a 1x20 pin array. For a current schematic and board layout see Appendix 3, 7.3.3.

The battery management module is comprised of several regulators, a small watch battery and a microcontroller. The input to the system is 12V DC provided by the main battery and power generation module. Outputs from the system include 5V, 3.3V and 1.8V. The 1.8V output goes directly to a pin on the “bone” to preserve the real-time clock. To create a “smarter circuit” a microcontroller, the ATMEGA168A, is implemented to shut down the regulators in the case of low power.

For example, if the the current battery in use is a 7.4V LiPo battery and our system continues to power as the battery goes below that nominal 7.4V, there is a possibility the battery will catch on fire and explode. To avoid this, the uC will take an ADC reading every 30 seconds from the battery. If the reading correlates to a voltage below the required level it will shut off the main regulators on the board that convert the 12V to 5V. The 5V outputs are responsible for powering the LCD screen and the “bone”. If the regulators are shut off the uC will wait 65 seconds to take another reading, hoping to find a voltage higher than before in the case of current recharge while pedalling. If not, the circuit will stay powered off until the nominal voltage level is attained. A schematic of the battery management module can be found on the attached schematic in the Appendix 3, 7.3.1, sheet 3 titled “Power Regulators and input”.

The LCD screen implemented is a 4” touch screen with refresh rate of 60Hz and 18-bit color. The screen is resistive touch and will talk directly with the “bone” via the main PCB board attached to the “bone” and a microcontroller. The touch response is 4-wire and takes in 5V for power and backlight. To change the backlight of the screen the 5V input can be scaled down, however, our system will utilize full backlight whenever powered to increase visibility in the sun. The screen will display a GUI comprising of several layouts. The first layout, a map screen, will contain a map with current location. To move maps the user can swipe left, right, up down, and a new map will be loaded. Much like the old video games, maps will switch based on the current location and, unless otherwise commanded, switch to a new “ttile” only when the current location is outside the bounds of the current block. Effectively the map layout will be in rectangular tiles that, when put together, resemble the entire loaded bitmap. The bitmaps will be stored on an external microSD card and loaded into an ATMEGA1284P microcontroller. This microcontroller will communicate with the “bone” via Two-Wire interface in order to update commands on current statistics and which tile should be on the current view. The second layout will contain user statistics including current heart rate, current speed, current distance travelled, time elapsed since reset/last clear. The third layout will have a data transfer screen that allows the user to clear the stored trip data on the “bone” as

well as transfer that information to a USB stick, via the “bone’s” USB port, in .csv or .txt format.

4.0 First Semester Progress

In the first semester, design has begun for each module, and implementation and testing have begun for the LCD Screen, BeagleBone and Peripheral Boards, Power Management, and Heart Rate modules. The parts for the Sensors and Power Generation modules have arrived and are ready for testing and implementation over break/early in the Spring semester, while the GUI is in design.

The LCD screen, as demonstrated in the First Deliverables Testing, has been tested on an Arduino board to show the capabilities of navigation between nine screens which will eventually contain various applications for the user. Additionally, it was demonstrated that we can maintain data corresponding to the screen positions, calibrate the screen to greater than 95% accuracy, and save these settings according to the user’s decision. Research concerning sunlight readability is ongoing.

Ability to communicate with and control the BeagleBone was also demonstrated in the First Deliverables Testing. The BeagleBone was shown to stream NMEA strings from an onboard GPS, and the NMEA string parser is under development in Python. The USB stream test was considered successful as NMEA strings appeared in the minicom terminal, indicating no errors from the GPS output to the USB input. Additionally, root access to the board and running C++ program with correct output to the terminal was demonstrated, along with GPIO functionality and UART pin implementations. The GPS module is considered complete.

The Power Management system has also been completed. As the final module shown in the First Deliverables Testing, this module has been shown to ring voltage from a LiPo 2-cell, and run an ADC conversion which triggers the main regulator to provide power to the board or send the microcontroller into a sleep state with the board, which shuts off when the voltage is less than 7.43V.

The Heart Rate sensor has been tested with an MSP430 with beats per minute output to a PC via UART over the USB available on the LaunchPad, although this UART data will eventually be sent to the BeagleBone in the final Bikeputer/Re.Cycle product.

5.0 Technical Plan

Task 1. PCB Design and fabrication

Todd is adding the finishing touches to the first revision of our main board. This is designed to mount several microcontrollers, and will establish communications between the controllers and external devices through peripherals. The LCD breakout board can also be directly mounted onto a header included on the board. This will be the first revision fabricated, so it will require substantial testing in order to establish what, if any, changes need be made.

Task 2. General module integration

To date, our design approach has consisted of a divide-and-conquer methodology. Each group member has been working in parallel toward completing personal responsibilities; this has allowed for accelerated completion of individual aspects of our system. As significant progress has been made, we feel it is appropriate to begin redirecting some of our efforts into integrating the sensors, main processor, and display together. This effort involves each member of the group, and will consist of heavy software development and moderate hardware in order to establish communication protocols and scheduling routines. Although the final implementation will be on the main board, it is difficult to predict precisely when the board will be available for population, so it is important that we prototype communications separately.

Task 3. Sensor control

Finalized control circuitry must be designed for each of the various sensors. For those which an interface is sufficiently developed, the emphasis is shifting toward development of compatibility. To clarify, these modules have been designed as stand-alone implementations, and now must be altered to retain functionality but to operate under commands received from the ARM processor. This encompasses system specific alterations for each group member, as well as development of software compatible with our embedded Linux environment.

Task 4. Lab testing

We have established that it would be beneficial to be able to provide quantitative data corresponding to the power output from our generator. Lab testing will include attempts to make correlative measures between RPM and voltage output, as well as measurements of power production from live tests of stationary cycling. This will give us a better idea of how much power we will need the battery to provide, as well as how flexible our power budget truly is.

Task 5. Power system integration

This consists of combining the battery, power management system, and generator together into a functional unit. Upon completion, we will be able to test how long our system can run with variable support from the generator, or purely on battery power. Another variable for testing is how long our system can run with different combinations of features utilized.

Task 6. Component version control

After these previous tasks are complete, progress will be sufficient enough that we each need to confirm that the components used in our system are finalized. For example, significant development has been made for the LCD touchscreen and its associated controller, however the final version of our project may incorporate a significantly larger screen.

Task 7. Physical enclosure

One of the last tasks we will take on is the creation of a method for conveniently attaching each component to the bicycle that has been provided for us. We hope also to have some form of contemporary encasing, potentially even weather proof if time permits.

6.0 Budget Estimate

Module	Cost ($)
BeagleBone	89.00
GPS	40.00
Battery Management	● ADP2302- 3.00 each (6.00 total) ● MCP1703- 0.68 each (1.36 total) ● Atmega168A- 2.43 9.79 total
Power Generator	40.00
Heart Rate	● Sensor- 25.00 ● ATtiny84A- 1.35 26.35 total
Speed Sensors	● Reed switch + magnet- 5.00 ● ATtiny84A- 1.35 6.35 total

LCD Screen	• Screen- 40.00 • Atmega328- 5.00 45.00 total
Test Bicycle	Donated
Casing and Fabrication	Expected: 50.00
Total:	306.49

Thus far, the dominating item is the BeagleBone, followed by the GPS, Power Generator, and LCD Screen. The unknown costs are the casing and fabrication, which are nevertheless not worrisome considering how far below the budget the final product is expected to cost. The worst-case scenario emergency costs would include the need for a different screen or new BeagleBone, but there is plenty of space in the budget to take care of such issues, should they arise.

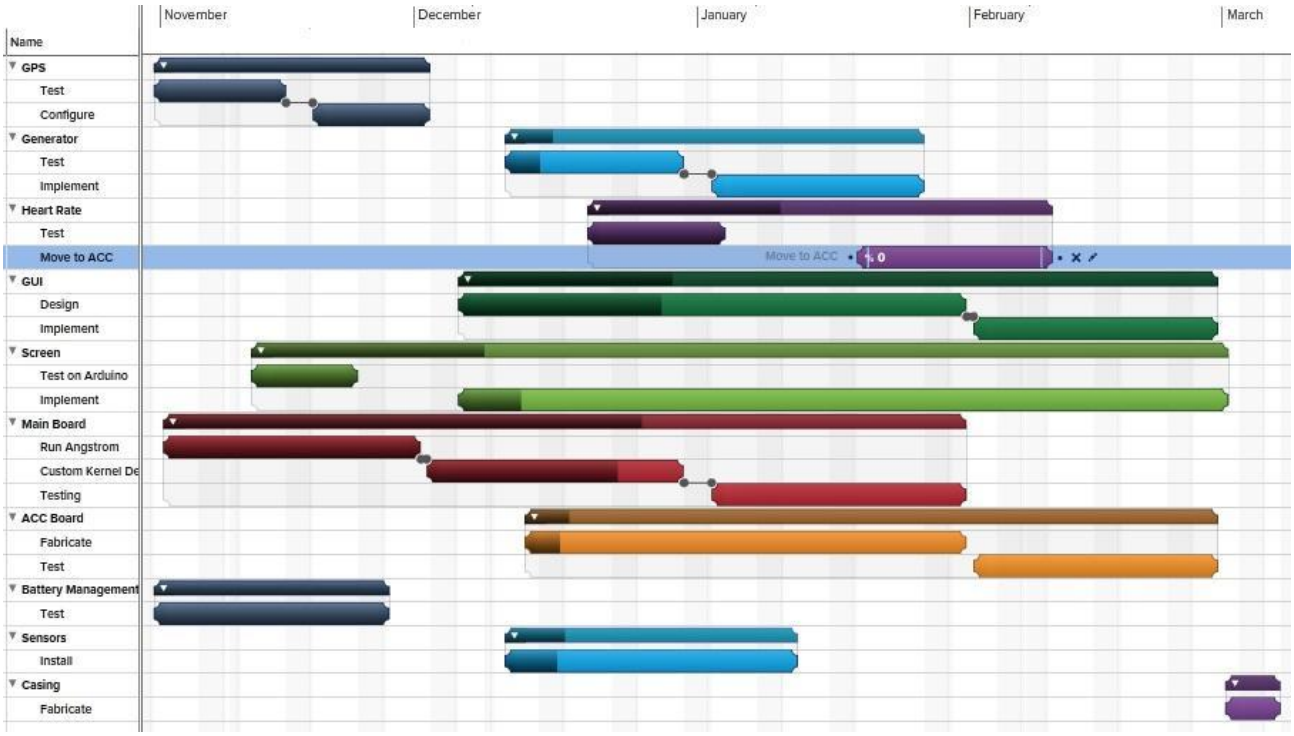
7.0 Attachments

7.1 Appendix 1- Engineering Requirements

Casing Dimensions	4.5”x4.5”x2”
Portability	All components detachable from bicycle within two minutes
Speed Error	<.25mph
Heart Rate Error	Within 4 bpm
GPS integration	Accurate location within three meters
User Storage	Ride length in time, distance, average speed, average HR.
Storage Accessibility	USB dumpable in .csv/.txt file, clearable from module
Rechargeable Pedal Power	Wall chargeable within twelve hours; pedal power supplies half of current needed @ 12V DC

Resistive Touch-Screen	Minimum 4” LCD panel refreshed at 60Hz, 18bit color. Sunlight Readable
GUI	Multitasking between heart rate, GPS, speed/distance.
Safety Constraint	Default clock screen when travelling > 4mph
Real-Time Clock Backup	RTC preserved for minimum of 5 days without main battery
Power Management	Stable 3.3V, 5V, 1.8V outputs for circuit board components

7.2 Appendix 2- Gantt Chart

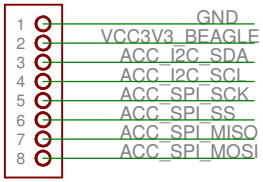


7.3 Appendix 3- Attached Documents

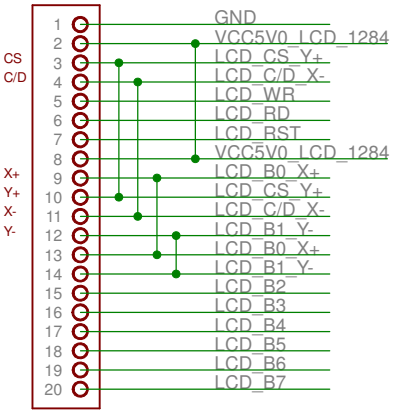
7.3.1- Current Board Design (Produce with “Eagle PCB”)

This is the latest schematic and board layout for the main PCB module. Not there are two peripheral board headers: One for the HR/Speed board and another for the GPS board. The board connects to the Beagle Bone’s main headers, providing power and communication.

Peripheral Breakout Connector- Goes directly to BB

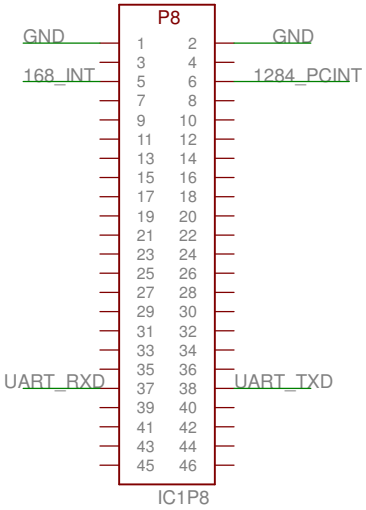


LCD Header



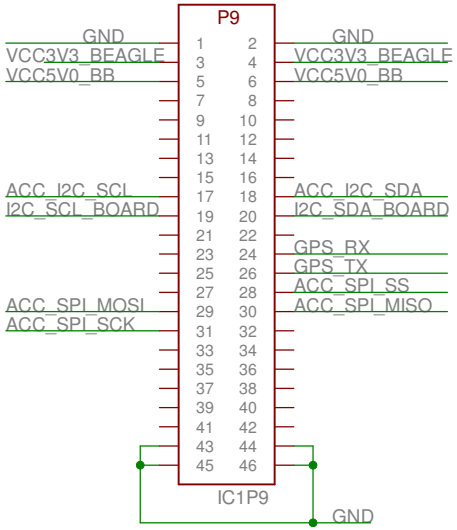
Note: This follows the Arduino Uno synchonization with LCD. 4-Wire interface

BeagleBone P8 Header



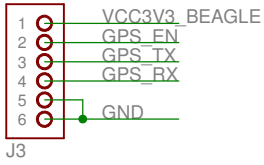
NOTE: Mode 4 for pins 37/38 -> UART
Mode 7 for GPIO's

BeagleBone P9 Header



NOTE: Pins 11 and 13 in mode 6
ACC_SPI pins mode 3
Pins 21/22 mode 1
Pins 17/18 mode 2
Pins 11/13 mode 6

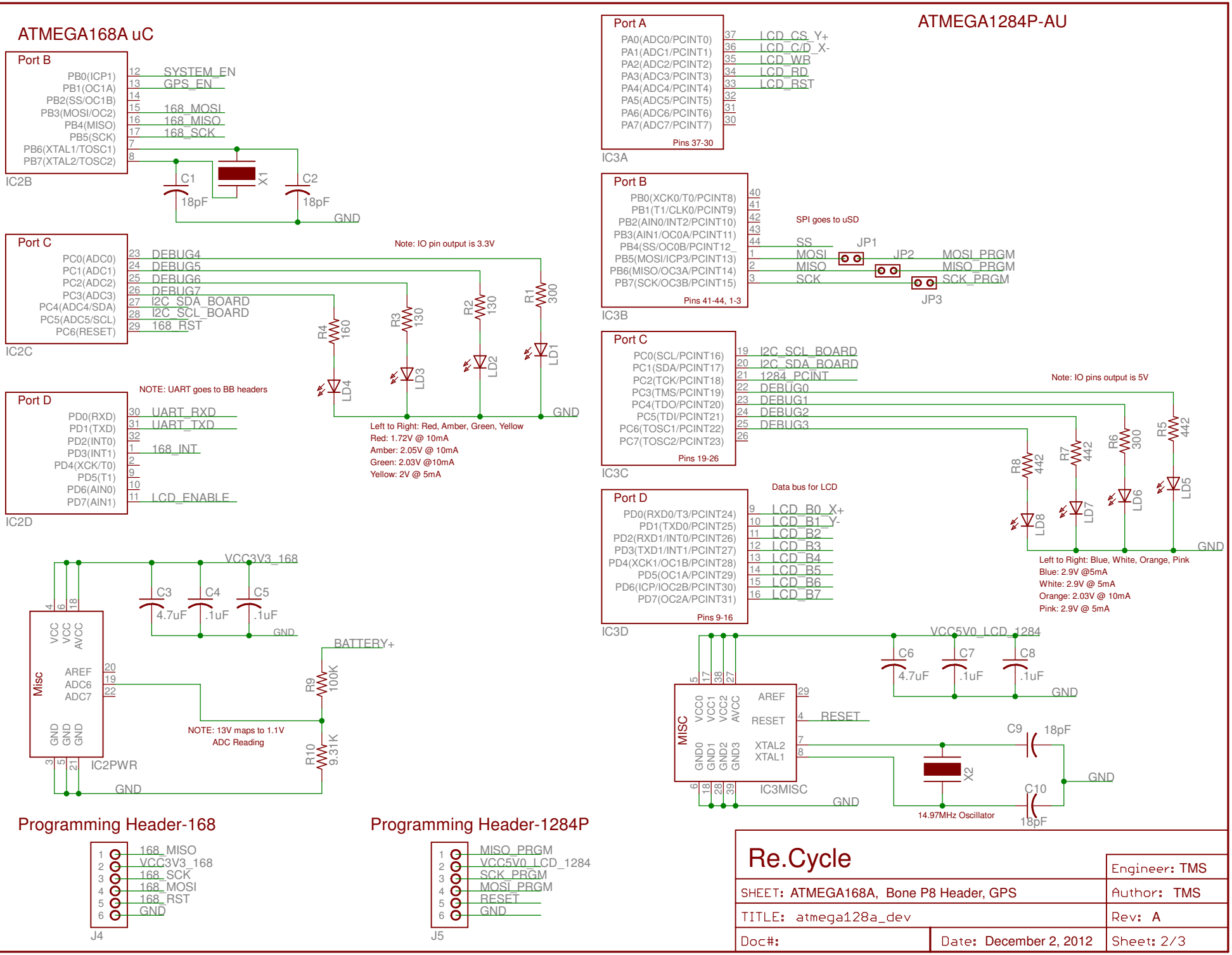
GPS Breakout Header



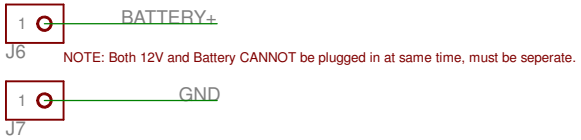
NOTE: This is on breakout board, should point towards sky
GPS UART goes directly to BB Headers

BeagleBone will talk with 128 via UART1, processing which map is needed and so forth. The 128 communicates via SPI with the microSD card.
BeagleBone communicates with the 168 via UART, receiving data on Battery Voltage and NMEA strings both via UART.

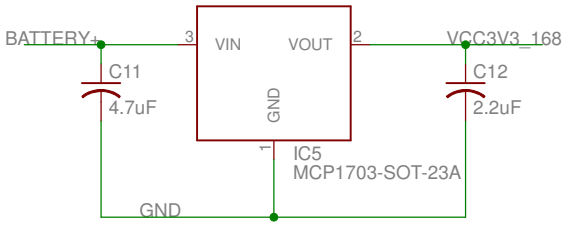
Re.Cycle		Engineer: TMS
SHEET: BB, LCD, Peripheral, GPS Headers		Author: TMS
TITLE: atmega128a_dev		Rev: A
Doc#: 1	Date: December 2, 2012	Sheet: 1/3


ATMEGA168A uC
Port B
PB0(ICP1)
PB1(OC1A)
PB2(SS/OC1B)
PB3(MOSI/OC2)
PB4(MISO)
PB5(SCK)
PB6(XTAL1/TOSC1)
PB7(XTAL2/TOSC2)
12 SYSTEM_EN
13 GPS_EN
14
15 168_MOSI
16 168_MISO
17 168_SCK
7
8
IC2B
C1
18pF
X1
C2
18pF
GND
Port C
PC0(ADC0)
PC1(ADC1)
PC2(ADC2)
PC3(ADC3)
PC4(ADC4/SDA)
PC5(ADC5/SCL)
PC6(RESET)
23 DEBUG4
24 DEBUG5
25 DEBUG6
26 DEBUG7
27 I2C_SDA_BOARD
28 I2C_SCL_BOARD
29 168_RST
IC2C
Note: IO pin output is 3.3V
R4 160
R3 130
R2 130
R1 300
LD4
LD3
LD2
LD1
GND
Left to Right: Red, Amber, Green, Yellow
Red: 1.72V @ 10mA
Amber: 2.05V @ 10mA
Green: 2.03V @10mA
Yellow: 2V @ 5mA
Port D
PD0(RXD)
PD1(TXD)
PD2(INT0)
PD3(INT1)
PD4(XCK/T0)
PD5(T1)
PD6(AIN0)
PD7(AIN1)
30 UART_RXD
31 UART_TXD
32
1 168_INT
2
9
10
11 LCD_ENABLE
IC2D
NOTE: UART goes to BB headers
VCC3V3_168
C3 4.7uF
C4 .1uF
C5 .1uF
GND
Misc
VCC
VCC
AVCC
AREF
ADC6
ADC7
GND
GND
GND
IC2PWR
BATTERY+
R9 100K
R10 9.31K
NOTE: 13V maps to 1.1V
ADC Reading
GND
Programming Header-168
1 168_MISO
2 VCC3V3_168
3 168_SCK
4 168_MOSI
5 168_RST
6 GND
J4
Programming Header-1284P
1 MISO_PRGM
2 VCC5V0_LCD_1284
3 SCK_PRGM
4 MOSI_PRGM
5 RESET
6 GND
J5
ATMEGA1284P-AU
Port A
PA0(ADC0/PCINT0)
PA1(ADC1/PCINT1)
PA2(ADC2/PCINT2)
PA3(ADC3/PCINT3)
PA4(ADC4/PCINT4)
PA5(ADC5/PCINT5)
PA6(ADC6/PCINT6)
PA7(ADC7/PCINT7)
Pins 37-30
37 LCD_CS_Y+
36 LCD_C/D_X-
35 LCD_WR
34 LCD_RD
33 LCD_RST
32
31
30
IC3A
Port B
PB0(XCK0/T0/PCINT8)
PB1(T1/CLK0/PCINT9)
PB2(AIN0/INT2/PCINT10)
PB3(AIN1/OC0A/PCINT11)
PB4(SS/OC0B/PCINT12_
PB5(MOSI/ICP3/PCINT13)
PB6(MISO/OC3A/PCINT14)
PB7(SCK/OC3B/PCINT15)
Pins 41-44, 1-3
40
41
42
43
44 SS
1 MOSI
2 MISO
3 SCK
SPI goes to uSD
JP1
JP2
JP3
MOSI_PRGM
MISO_PRGM
SCK_PRGM
IC3B
Port C
PC0(SCL/PCINT16)
PC1(SDA/PCINT17)
PC2(TCK/PCINT18)
PC3(TMS/PCINT19)
PC4(TDO/PCINT20)
PC5(TDI/PCINT21)
PC6(TOSC1/PCINT22)
PC7(TOSC2/PCINT23)
Pins 19-26
19 I2C_SCL_BOARD
20 I2C_SDA_BOARD
21 1284_PCINT
22 DEBUG0
23 DEBUG1
24 DEBUG2
25 DEBUG3
26
IC3C
Note: IO pins output is 5V
R8 442
R7 442
R6 300
R5 442
LD8
LD7
LD6
LD5
GND
Left to Right: Blue, White, Orange, Pink
Blue: 2.9V @5mA
White: 2.9V @ 5mA
Orange: 2.03V @ 10mA
Pink: 2.9V @ 5mA
Port D
PD0(RXD0/T3/PCINT24)
PD1(TXD0/PCINT25)
PD2(RXD1/INT0/PCINT26)
PD3(TXD1/INT1/PCINT27)
PD4(XCK1/OC1B/PCINT28)
PD5(OC1A/PCINT29)
PD6(ICP/IOC2B/PCINT30)
PD7(OC2A/PCINT31)
Pins 9-16
9 LCD_B0_X+
10 LCD_B1_Y-
11 LCD_B2
12 LCD_B3
13 LCD_B4
14 LCD_B5
15 LCD_B6
16 LCD_B7
IC3D
Data bus for LCD
VCC5V0_LCD_1284
C6 4.7uF
C7 .1uF
C8 .1uF
GND
MISC
VCC0
VCC1
VCC2
AVCC
GND0
GND1
GND2
GND3
AREF
RESET
XTAL2
XTAL1
RESET
C9 18pF
X2
C10 18pF
14.97MHz Oscillator
GND
IC3MISC
GND
Re.Cycle
Engineer: TMS
SHEET: ATMEGA168A, Bone P8 Header, GPS
Author: TMS
TITLE: atmega128a_dev
Rev: A
Doc#:
Date: December 2, 2012
Sheet: 2/3

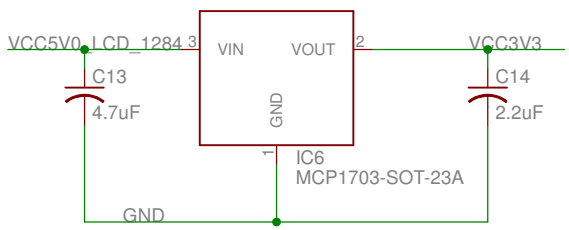
Battery Power



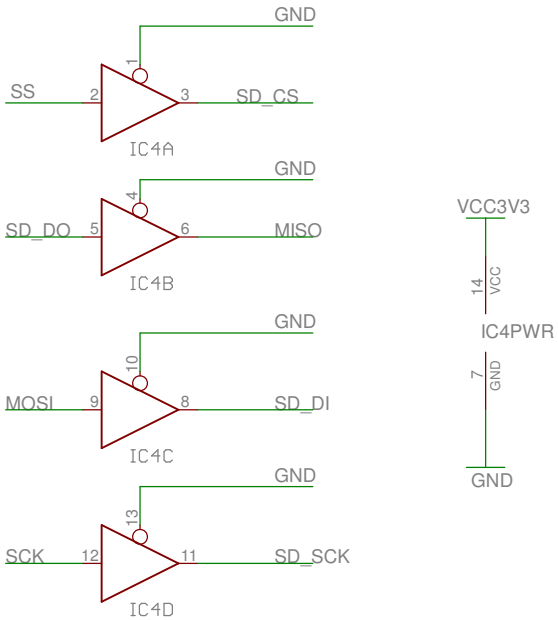
5v0 to 3v3 regulator for AT168



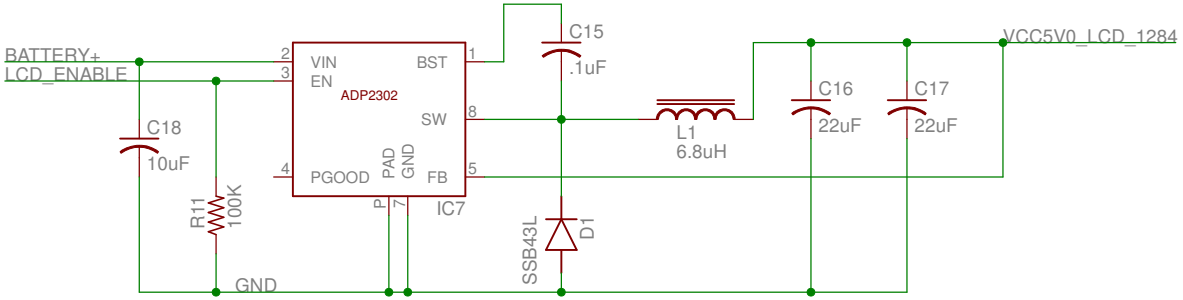
Atmega128 5V->3.3V Regulator



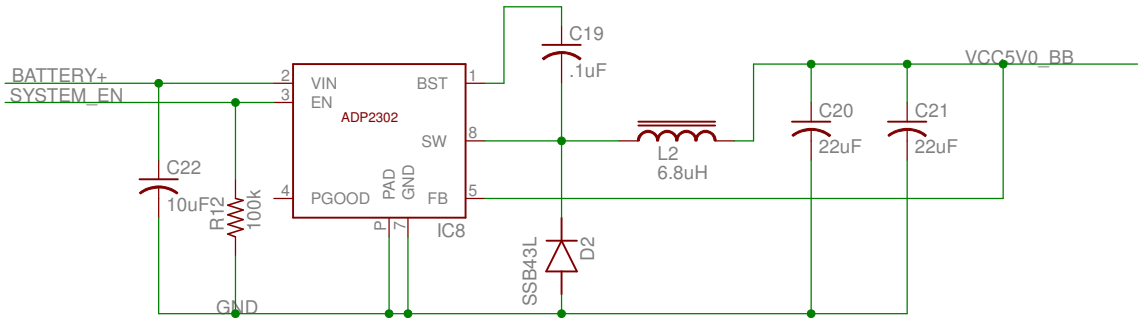
SD Breakout Buffer Array



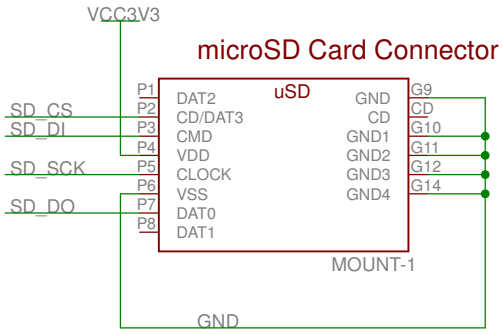
12V->5V Main Regulator



ATMEGA128 and BB 12V->5V Regulator

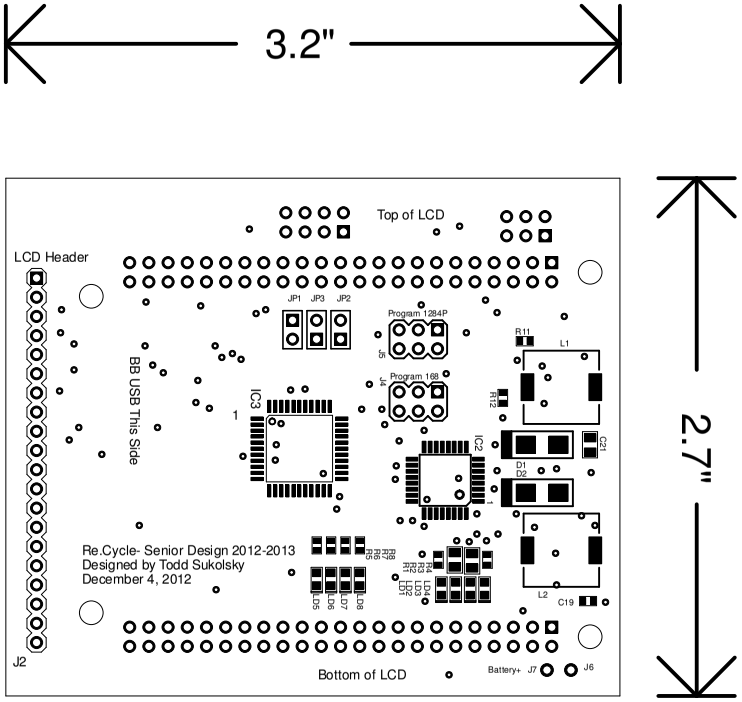


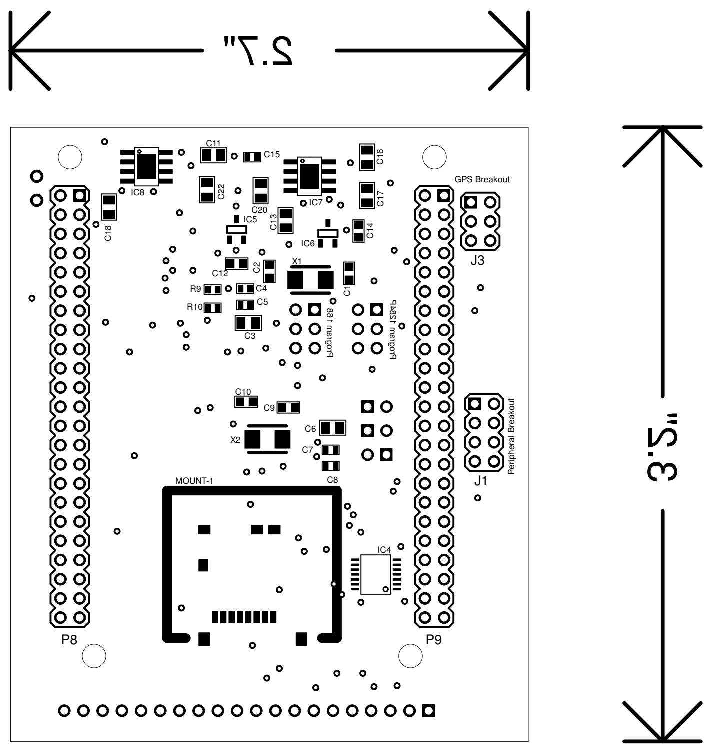
microSD Card Connector



Note: No chip detect, assume it's in correctly if transmitting

Re.Cycle		Engineer: TMS
SHEET: Power Regulators and Input		Author: TMS
TITLE: atmega128a_dev		Rev: A
Doc#:	Date: November 23, 2012	Sheet: 3/3


3.2"
2.7"
Top of LCD
LCD Header
JP1
JP3
JP2
Program 1284P
Program 168
R11
L1
R12
C21
IC3
IC2
D1
D2
BB USB This Side
Re.Cycle- Senior Design 2012-2013
Designed by Todd Sukolsky
December 4, 2012
L2
C19
J2
Bottom of LCD
Battery+
J7
J6


2.7"
3.5"
GPS Breakout
J3
Peripheral Breakout
J1
P8
P9
MOUNT-1
IC4
IC5
IC6
IC7
IC8
X1
X2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
C14
C15
C16
C17
C18
C20
C22
R9
R10