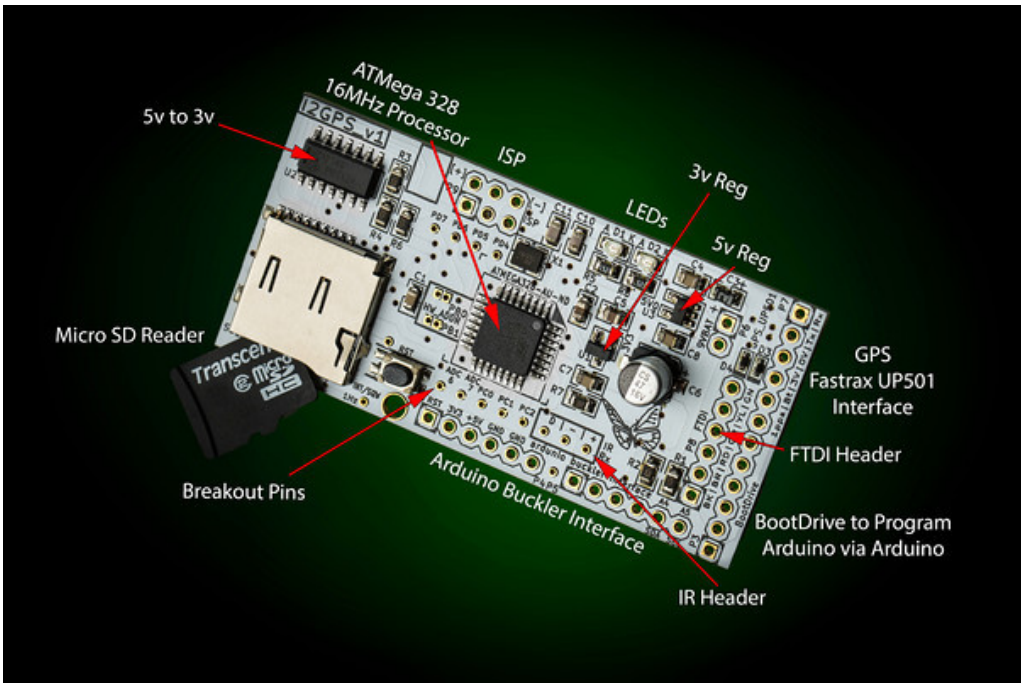


I2GPS v1

An easy to use RTC, GPS and SD memory card interface

brought to you by

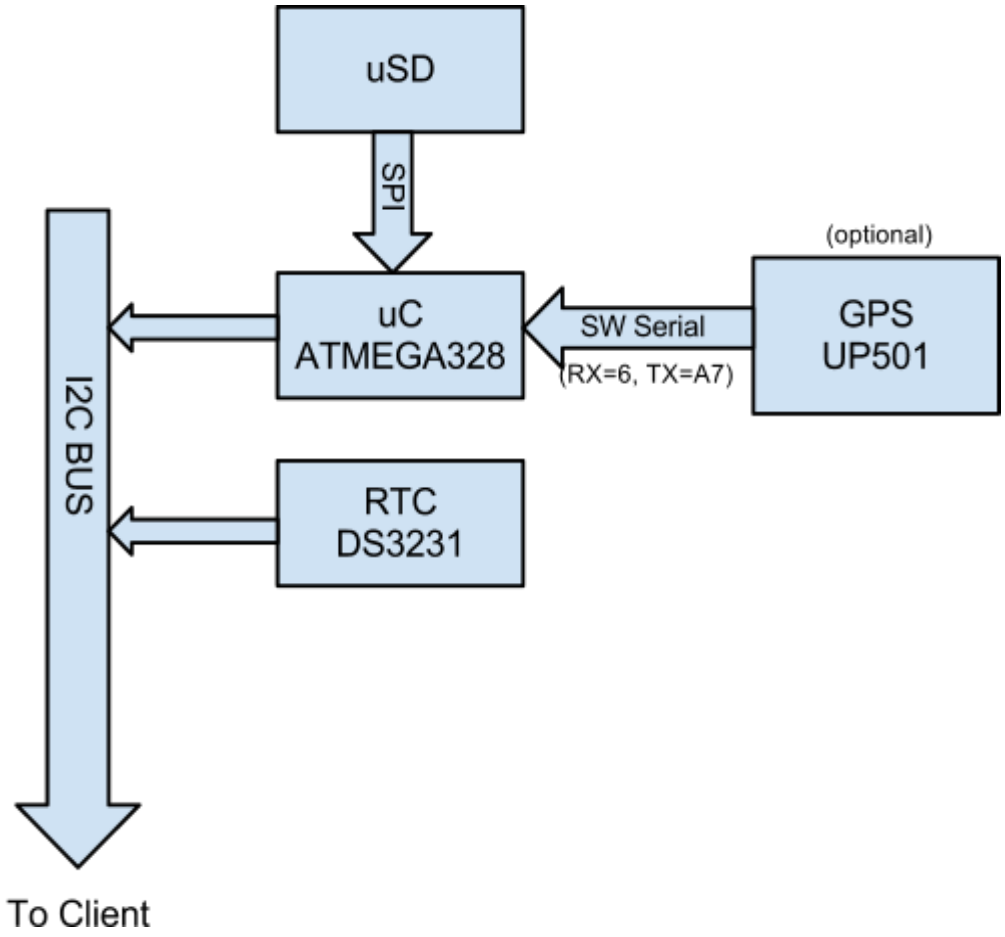
Features

micro SD card reader
Temperature controlled, precision Real Time clock, with battery backup
GPS interface for the Fastrax UP501 module
Arduino compatible
ChronoDot compatible
Two general purpose LEDs
IR interface
I2C interface requires no additional pins when used on an existing I2C bus
Unused I/O pins are available via convenient breakout pads.

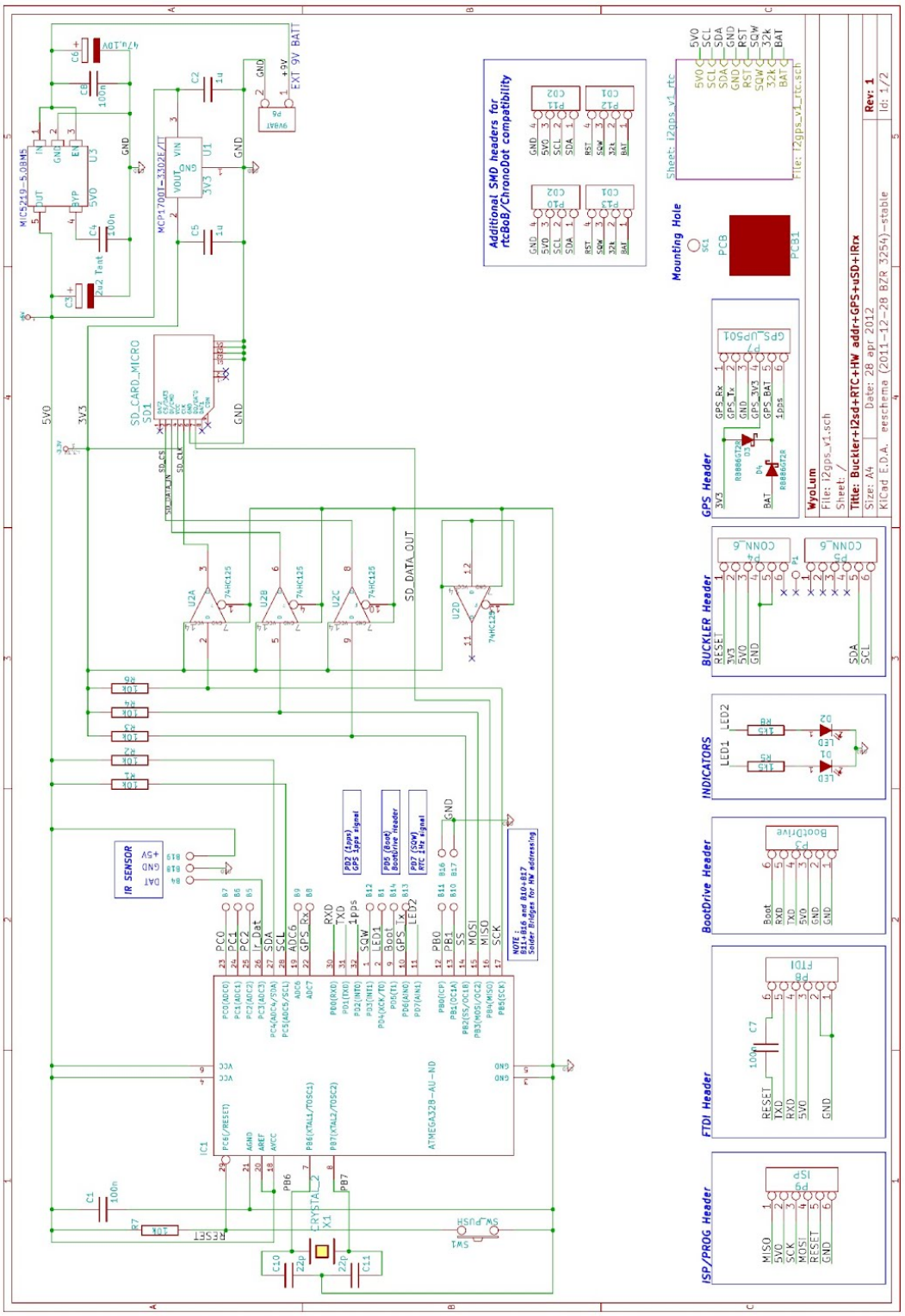
Potential Uses

- Stand-alone data logger
- Simple-to-use, persistent storage
- Program loader for separate Arduino compatible
- Store events, animations, and sounds for ClockTHREE
- Automatically set the time and adjust time zones - for ClockTHREE and ClockTHREEjr

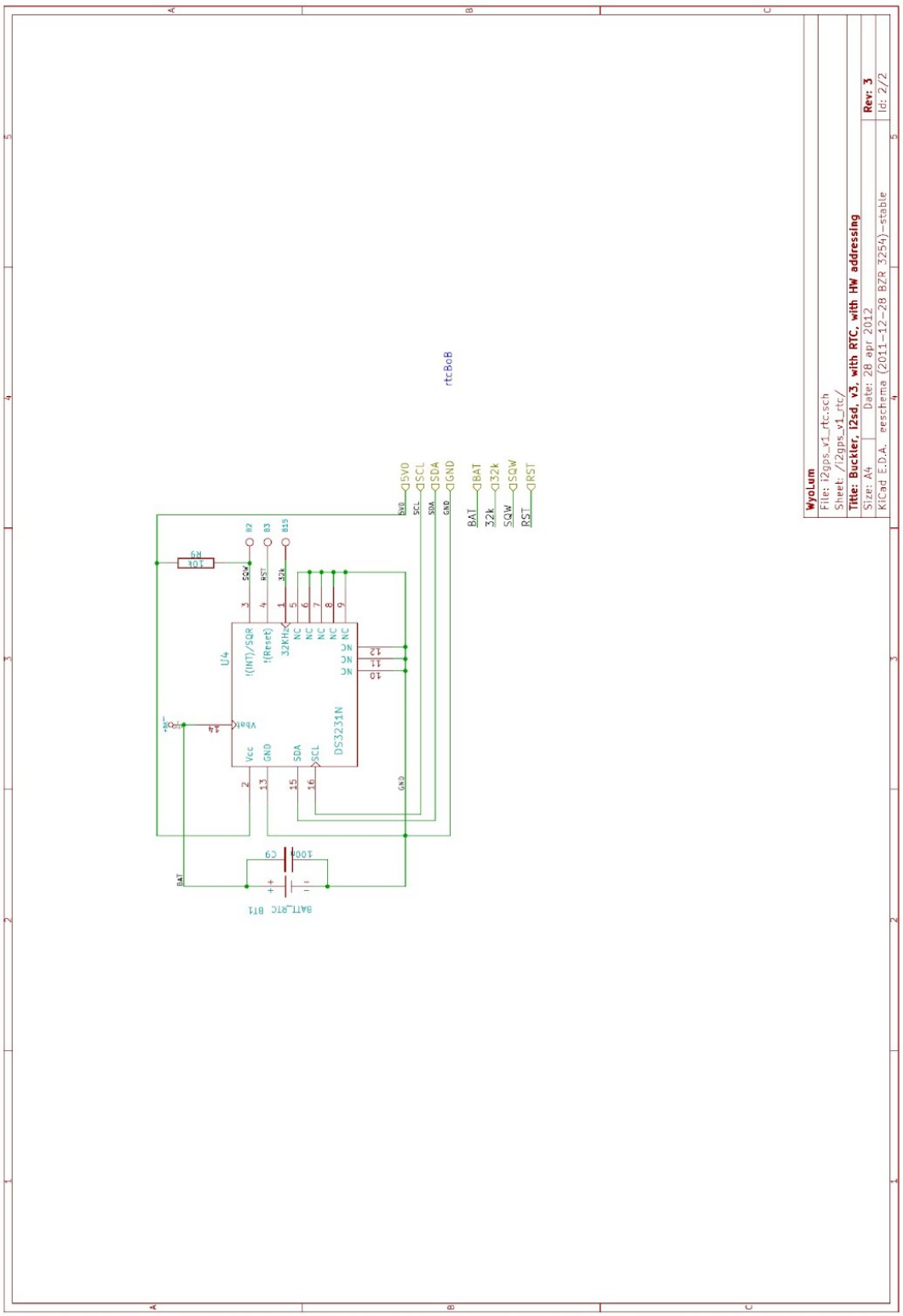
Block Diagram



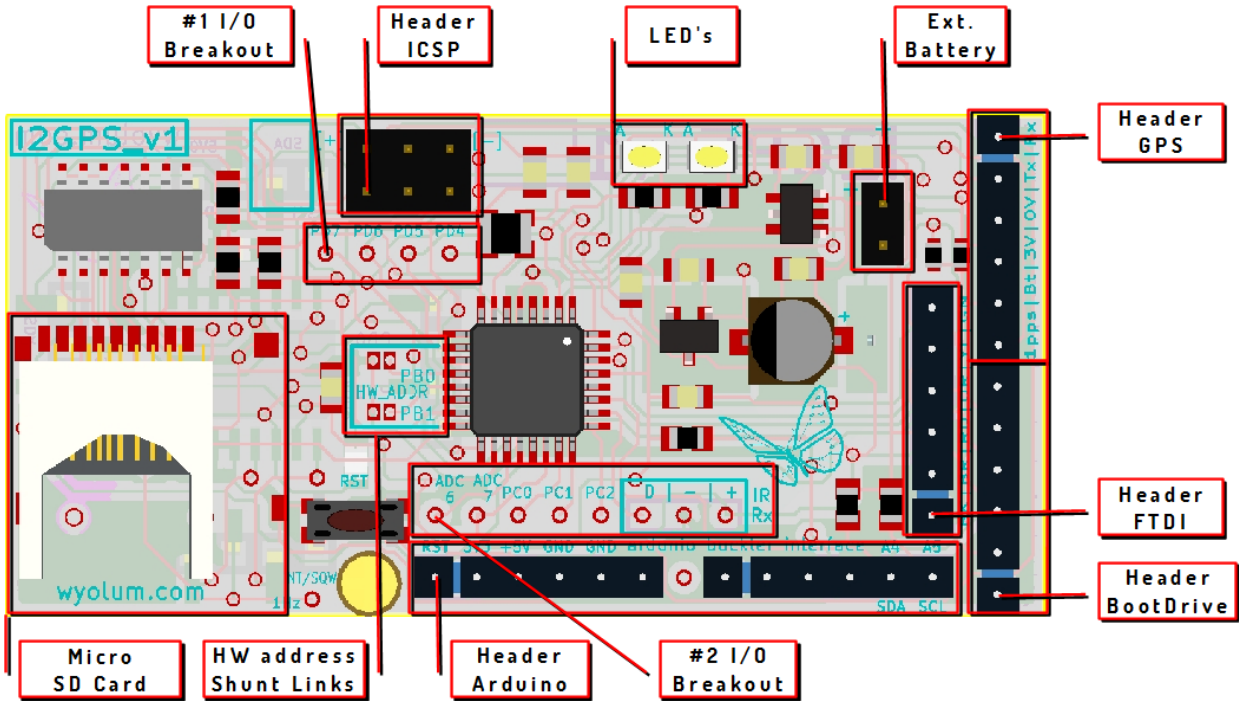
Schematic, #1

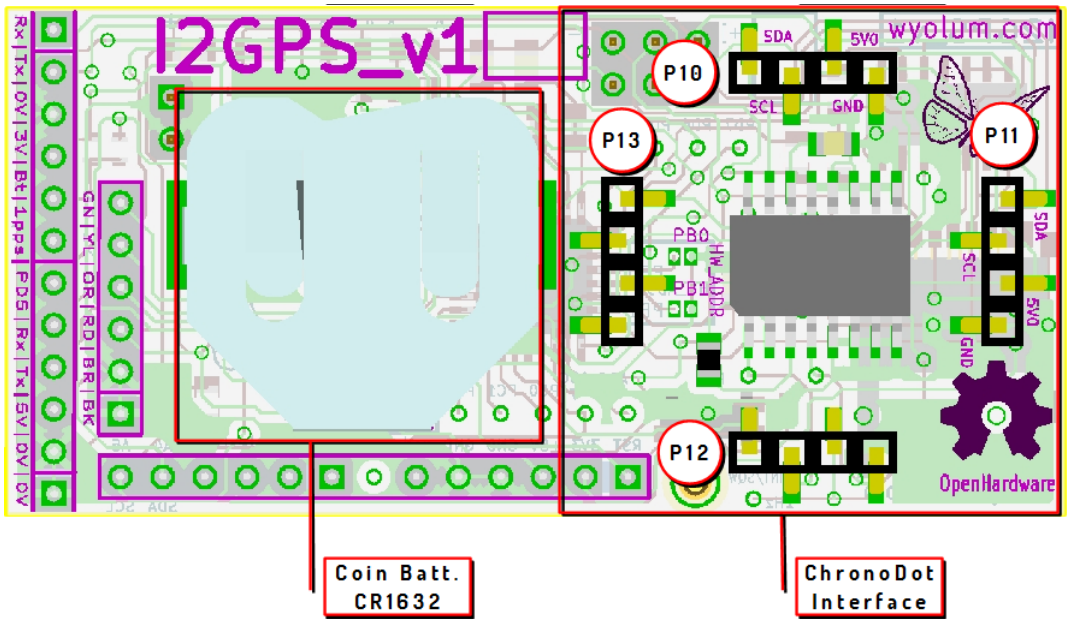

Additional SMD headers for rtcBoB/ChronoDot compatibility
Mounting Hole
SCL
PCB
PCB1
Sheet: i2gps_v1_rtc
File: i2gps_v1_rtc.sch
5V0
SCL
SDA
GND
RST
SQW
32k
BAT
GPS Header
BUCKLER Header
INDICATORS
BootDrive Header
FTDI Header
ISP/PROG Header
IR SENSOR
SD_CARD_MICRO
SD_DATA_OUT
ATMEGA328-AU-ND
CRYSTAL_2
Wyo.Lum
File: i2gps_v1.sch
Sheet: /
Title: Buckler+i2sd+RTC+HW addr+GPS+uSD+IRrx
Size: A4
Date: 28 apr 2012
KiCad E.D.A. eeschema (2011-12-28 BZR 3254)-stable
Rev: 1
Id: 1/2

Schematic, #2


rtcBoB
5V0
SCL
SDA
GND
BAT
32k
SQW
RST
WyoLum
File: i2gps_v1_rtc.sch
Sheet: /i2gps_v1_rtc/
Title: Buckler, i2sd_v3, with RTC, with HW addressing
Size: A4
Date: 28 apr 2012
Rev: 3
KiCad E.D.A. eeschema (2011-12-28 BZR 3254)-stable
Id: 2/2
DS3231N
BATT_RTC
C9
100n

Physical Interfaces





Physical Interfaces, Description

(**RED** Markers point to Pin # 1 of each header)

 # 1 I/O Breakout

1 = PD7 (Arduino digital pin 7)
2 = PD6 (Arduino digital pin 6)
3 = PD5 (Arduino digital pin 5)
4 = PD4 (Arduino digital pin 4)

Header ICSP

1 = MISO
2 = 5V0
3 = SCK
4 = MOSI
5 = RESET
6 = GND

LED's

D1 = PD4 (Arduino digital pin 4, Active HIGH)
D2 = PD7 (Arduino digital pin 7, Active HIGH)

EXT BATT

1 = + 9 V max. +5.5V Min
2 = GND +0V

HEADER GPS

Note: to use the SoftwareSerial libary:

```
#include "SoftwareSerial.h"
…
SoftwareSerial sws(6, A7);
```

1 = GPS_Rx , ADC7 (Arduino analog pin 7, A7)
2 = GPS_Tx , PD6 (Arduino digital pin 6)
3 = GND
4 = 3V3
5 = GPS_BATT
6 = 1pps , PD2

HEADER BOOTDRIVE

1 = GND
2 = GND
3 = 5V0
4 = TXD , PD1
5 = RXD , PD0
6 = Boot , PD5

HEADER FTDI

1 = GND
2 = GND
3 = 5V0
4 = RXD , PD0
5 = TXD , PD1
6 = RESET

HEADER ARDUINO (Buckler)

1 = RESET (Connected to I2GPS reset and Host reset so that both reset at the same time. disconnect for independent resets)
2 = 3V3
3 = 5V0
4 = GND
5 = GND
6 = NC
7 = NC
8 = NC
9 = NC
10 = NC
11 = NC
12 = PC4, SDA (I2C interface)
13 = PC5, SCA (I2C interface)

 # 2 I/O BREAKOUT

1 = ADC6
2 = ADC7 , GPS_Rx
3 = PC0
4 = PC1
5 = PC2
6 = Ir_Dat , PC3
7 = GND
8 = 5V0

NOTE : 6 , 7 , 8 for Ir Receiver

HW ADDRESS SHUNT LINKS

PB0 (Arduino Pin 8)
PB1 (Arduino Pin 9)

NOTE : These pins are intended to be used for changing I2C address. Default program uses 88 as the base address (although this is completely arbitrary as long as the address does not conflict with any other devices on your I2C bus.

NOTE: Use ATMEGA internal pull-ups on these pins. For instance, to set internal pull-up on pin 8 use this code snippet:

```
pinMode(8, INPUT);
digitalWrite(8, HIGH);
```

PB0 (Arduino Pin 8)	PB1 (Arduino Pin 9)	I2C Address
LOW	LOW	88
LOW	HIGH	89
HIGH	LOW	90
HIGH	HIGH	91

To change I2C Address, jumper one or both of these ports to GND

MICRO SD-CARD

Note

SD_CS = PB2 , SS
SD_DATA_IN = PB3 , MOSI
SC_CLK = PB5 , SCK
SD_DATA_OUT = PB4 , MISO

CHRONODOT INTERFACE

P10 , # 1 = SDA
P10 , # 2 = SCL
P10 , # 3 = 5V0
P10 , # 4 = GND

P11 , # 1 = SDA
P11 , # 2 = SCL
P11 , # 3 = 5V0
P11 , # 4 = GND

P12 , # 1 = BAT (NOT CONNECTED)
P12 , # 2 = 32k
P12 , # 3 = SQW
P12 , # 4 = RST

P13 , # 1 = BAT
P13 , # 2 = 32k (NOT CONNECTED)
P13 , # 3 = SQW
P13 , # 4 = RST (NOT CONNECTED)

NOTE

Ref. designators are placed near Pin #1.

Use wire jumpers (red wires) **if** any "NOT CONNECTED" pin is required.
The four connectors allow two possible mounting positions.
Lateral , using P11 and P13. Longitudinal , using P10 and P12

Software Interface Options

I2GPS (I2C interface to SD and GPS)

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Function	Format	LSB
0x00	DATETIME								UNIX Time	unsigned long	1 SECOND
0x01											
0x02											
0x03											
0x04	LATITUDE								Latitude	signed int	.001 DEGREE
0x05											
0x06											
0x07											
0x08	LONGITUDE								Longitude	signed int	.001 DEGREE
0x09											
0x0A											
0x0B											
0x0C	ALTITUDE								Altitude	signed int	1 CM
0x0D											
0x0E											
0x0F											
0x10	SPEED								Speed	unsigned long	.01 KNOT
0x11											
0x12											
0x13											
0x14	COARSE								Course	unsigned long	.01 DEGREE
0x15											
0x16											
0x17											
0x18	FIX AGE								Fix age	unsigned long	.001 SECOND
0x19											
0x1A											
0x1B											
0x1C	YEAR								Year (2 digit)	unsigned byte	1 YEAR
0x1D	MONTH								Month	unsinged byte	1 MONTH
0x1E	DAY								Day	unsigned byte	1 DAY
0x1F	HOUR								Hour	unsinged byte	1 HOUR
0x20	MINUTE								Minute	unsigned byte	1 MINUTE
0x21	SECOND								Second	unsinged byte	1 SECOND
0x22	RATE								Log Rate	unsigned int	SECOND
0x23											
0x24	A0	A1	A2	A3	X	X	A6	A7	Analog Pin Logging	boolean	NA
0x25									Reserved		
0x26	D0	D1	D2	D3	D4	D5	D6	D7	Digital Pin Direction	bitfield	0=INPUT/1=OUTPUT
0x27	D0	D1	D2	D3	D4	D5	D6	D7	Digital Pin Read/Write	bitfield	
0x28	D0	D1	D2	D3	D4	D5	D6	D7	Digital Pin Logging	bitfield	
0x29									Reserved		
0x2A	pwm_value								D3 PWM WRITE	byte	
0x2B	pwm_value								D5 PWM WRITE	byte	
0x2C	pwm_value								D6 PWM WRITE	byte	
0x2D	pwm_value								D9 PWM WRITE	byte	
0x2E	pwm_value								P10 PWM WRITE	byte	
0x2F	FILENAME (DOS 8.3)								SD FILENAME	char[12]	
0x30											
0x31											
0x41							ENABLE	WRITE	SD File Flags	byte	
0x42	TELL/SEEK								File Position	unsigned long	
0x43											
0x44											
0x45											
0x46	SD FILE DATA								Read/Write data	char[32]	
0x47											
...											
0x67	ERROR CODE								Error code	byte	

GPS INTERFACE

This interface provides time, position, and motion information when the GPS receiver is tracking without a large processing or memory burden on the host processor. The data is read from a large table stored on the I2GPS depicted above. To read from the table, send one byte of data to the I2GPS to set the starting address. Then request the desired number of bytes to read. For instance, to read latitude and longitude, start at address 0x04 and read 8 bytes. You can also log GPS data to a file open for writing. (See SD interface). Some helper functions are defined I2GPS.h. See interface below.

```
/*
 * Read n_byte from slave starting from offset address addr.
 * Store result stored in dest (which must be at least n_byte long).
 * Return true if successful.
 *
 * Must be preceded with call to Wire.begin()
 */
void gps_raw_write(uint8_t addr, uint8_t n_byte, uint8_t *source);

/*
* write single byte
*/
   void gps_raw_write1(uint8_t addr, uint8_t data_byte);

/*
 * Read n_byte bytes from I2GPS starting at address addr.
 * Store result in dest.
 *
 * Must be preceded by a call to Wire.begin()
 */
bool gps_raw_read(uint8_t addr, uint8_t n_byte, uint8_t *dest);
```

Example GPS over I2C usage.

Client Code	Comment

`uint32_t lat;`
 	
`gps_raw_read(I2GPS_LAT_ADDR, 4, (uint8_t*)&lat);`
	Read latitude in .001 of a degree

`uint8_t gps_data[32];`
 	
`gps_raw_read(0, 32, gps_data);`
	Read 32 bytes of GPS data starting from address 0..

`uint8_t ymdhms[6];`
 	
`gps_raw_read(I2GPS_YEAR_ADDR, 6, ymdhms);`
	Read GPS year, month, day, hour minute, second.

SD INTERFACE

The SD interface works in very much the same way but it is a little more complicated because it adds the step of opening and closing a file. There can only be one file open at a time on the I2GPS.

Client Code	Comment

Example SD file access over I2C

LINKS

- website : www.wyolum.com
- e-mail : info@wyolum.com
- forum : http://wyolum.com/forum/forumdisplay.php?fid=12
- SVN Repo : http://code.google.com/p/clockthree/
- ChronoDot : http://macetech.com/store/index.php?main_page=product_info&cPath=5&products_id=8
- Arduino : http://www.arduino.cc/