GPS Bycicle Computer

Armin Schlegel
Christian Eismann

May 10, 2011

Overview

- Introduction
- The Toolchain
- Hardware
- Software
- Conclusion

The Project

Development of a small device which provides multiple information on an on-board display that are determined via the Global Positioning System

Main components:

- GPS receiver
- Graphic display
- SD card (recording of GPS data)

Tasks

- definition of features
- allocation of responsibilities
- assembling of a toolchain
- hardware development
- software development
- documentation

Hardware/Software development tools

Hardware:

- Eagle - circuit layout and design

Software:

- virtualbox (Oracle) including a Debian Linux image
- GNU Compiler Collection (AVR-GCC)
- avrdude (programmer)
- Make
- (sp)lint - static code analysis
- ISP Programmer

Software development tools

Makefile warning options:

Listing 1: Compiler warnings

```
# GCC compiler warnings
CWARN = -ffreestanding -pedantic -Wall -Wextra -Winit-self -Wswitch-default -
     Wunused-parameter -Wunknown-pragmas -Wstrict-overflow=1 -Warray-bounds -
     Wfloat-equal -Wdeclaration-after-statement -Wundef -Wno-endif-labels -
     Wshadow -Wbad-function-cast -Wcast-qual -Wcast-align -Wwrite-strings -
     Wstrict-prototypes -Wmissing-prototypes -Wmissing-declarations -Wredundant-
     decls -Wnested-externs -Wvla -Wvolatile-register-var -Wparentheses -g -Os -
     fno-strict-aliasing
```

Requirements

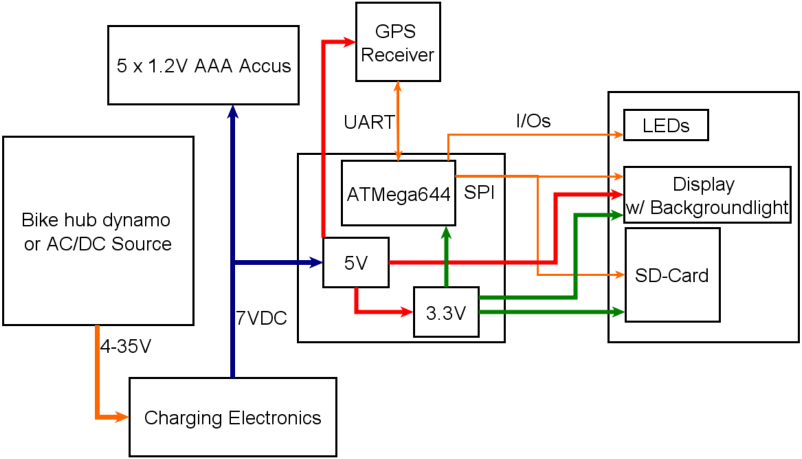
Basic, rudimentary Requirements:

- Processor supply voltage: 3.3V
- GPS receiver
- RS232 Debug/Communication Port (choosable via jumper)
- Usage of a graphic display
- SD card for data recording
- Mobile energy supply (chargeable)
- programmable via ISP (and JTAG)

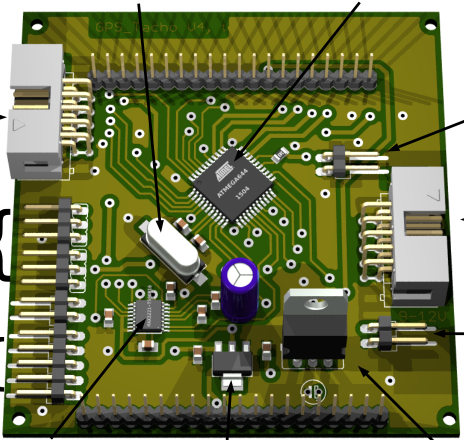
Main components

- ATmega32L and later ATmega644p (3.3V)
- NL-552ETTL (GPS Receiver, 5V VCC, 3.3V RXD/TXD levels)
- EA-DOGL128-6 (Display, 3.3V VCC, SPI)
- YAMAICHI SD slot (3.3V VCC, SPI)
- MAX3221 (RS232 controller, 3.3V VCC)

Block diagram



Layout: main board



7.3728 Mhz

Atmega644

JTAG

Reset Pins

ISP

RS232 Jumper

GPS-Con.

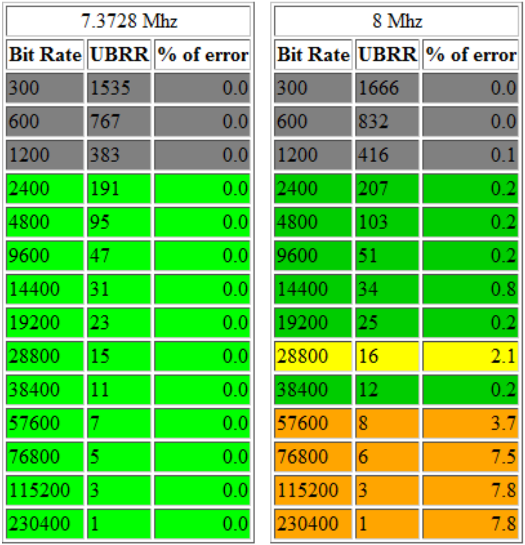
Input Voltage 6VDC-35VDC

MAX3221

5V-Regulator

3.3V-Regulator

Comparison between oscillators



7.3728 Mhz		
Bit Rate	**UBRR**	**% of error**
300	1535	0.0
600	767	0.0
1200	383	0.0
2400	191	0.0
4800	95	0.0
9600	47	0.0
14400	31	0.0
19200	23	0.0
28800	15	0.0
38400	11	0.0
57600	7	0.0
76800	5	0.0
115200	3	0.0
230400	1	0.0

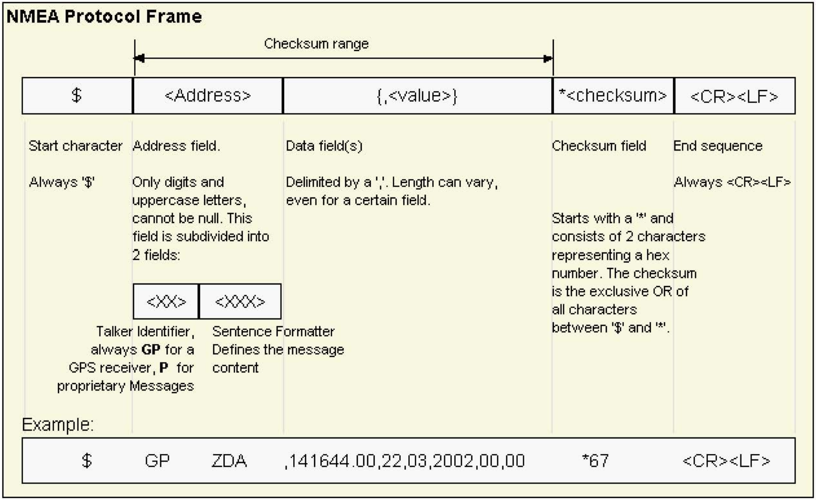
8 Mhz		
Bit Rate	**UBRR**	**% of error**
300	1666	0.0
600	832	0.0
1200	416	0.1
2400	207	0.2
4800	103	0.2
9600	51	0.2
14400	34	0.8
19200	25	0.2
28800	16	2.1
38400	12	0.2
57600	8	3.7
76800	6	7.5
115200	3	7.8
230400	1	7.8

Design

- mapping features to various modules
- SW running synchronous to GPS data receiving (USART interrupt)
- no usage of an OS

⇒ no timing/scheduling problems (general spoken: one big while(1) loop)

The NMEA/PUBX Protocol



NMEA Protocol Frame

Start character

Always '$'

Address field.

Only digits and uppercase letters, cannot be null. This field is subdivided into 2 fields:

Talker Identifier, always **GP** for a GPS receiver, **P** for proprietary Messages

Sentence Formatter Defines the message content

Data field(s)

Delimited by a ','. Length can vary, even for a certain field.

Checksum field

Starts with a '*' and consists of 2 characters representing a hex number. The checksum is the exclusive OR of all characters between '$' and '*'.

End sequence

Always <CR><LF>

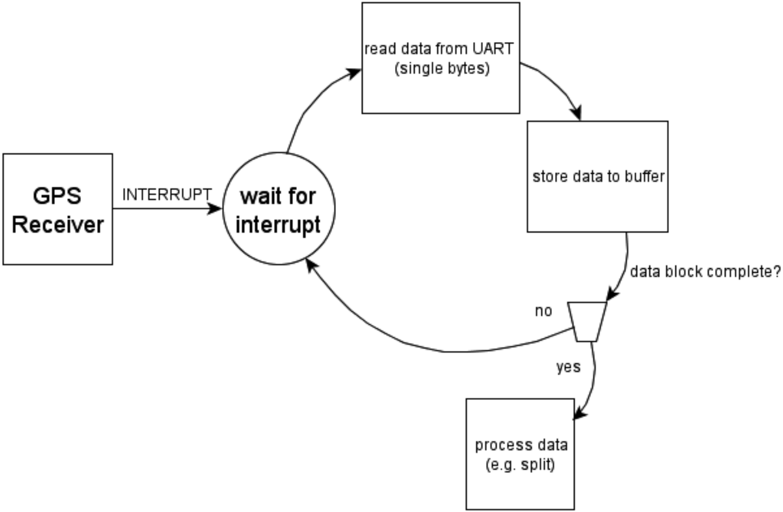
Example:

$	GP	ZDA	,141644.00,22,03,2002,00,00	*67	<CR><LF>

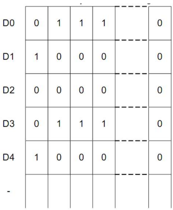
Components

- SPI (Serial Peripheral Interface)
    - Display
    - SDC/FAT16
- UART (Universal Asynchronous Receiver Transmitter)
    - GPS (Global Positioning System)
- touch screen functionality via ADC
- LEDs
- application

General workflow

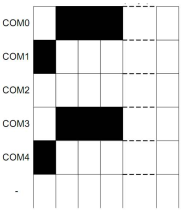


The graphic display



D0	0	1	1	1		0
D1	1	0	0	0		0
D2	0	0	0	0		0
D3	0	1	1	1		0
D4	1	0	0	0		0
-						

Display data RAM



Liquid crystal display

- internal RAM organized in pages
- each page contains 8 rows and 128 columns
- software solution: linear storage within an array

Display - Setting a single pixel

- direct pixel access via display data RAM
- virtual data representation is realized in linear order
- setting a pixel with coordinates X and Y:
  $\textbf{INDEX} = (Y * 8) + (X/8)$

Listing 2: Setting a single pixel

```c
void display_putpixel(unsigned char x, unsigned char y, uint8_t pixel_status)
{
   if (x < DISP_WIDTH && y < DISP_HEIGHT) {
      if (pixel_status == PIXEL_ON)
         disp_ram[(y >> 3) + (x << 3)] |= (1 << (y & 0x07));
      else
         disp_ram[(y >> 3) + (x << 3)] &= ˜(1 << (y & 0x07));
    }
}
```

GPS - Data storage

Listing 3: GPS data storage

```
#define GPS_RMC_BUFFER_SIZE 75

/* ... */

char * rmc[14] = { NULL };
static char gps_buffer_rmc[GPS_RMC_BUFFER_SIZE] = { '0' };

/* ... */

void gps_init_mem_rmc(void)
{
   rmc[0] = &(gps_buffer_rmc[0]);   /* $GPRMC */
   rmc[1] = &(gps_buffer_rmc[7]);   /* time: hhmmss.ssss */
   rmc[2] = &(gps_buffer_rmc[18]);  /* status */
   rmc[3] = &(gps_buffer_rmc[20]);  /* latitude: ddmm.mmmm */
   /* ... */
}
```

Problems

- ATmega32A RAM (2kB RAM too small)
- Windows line endings (\r\n) in NMEA format
- No provided SVN repository (used: Sourceforge)
- No provided bug/feature tracker

Project topics

- missing project environment
- very good support by Mr. Lenkowski
- underestimated effort for planned features (even by the Professor)

Ideas for future development

Hardware:

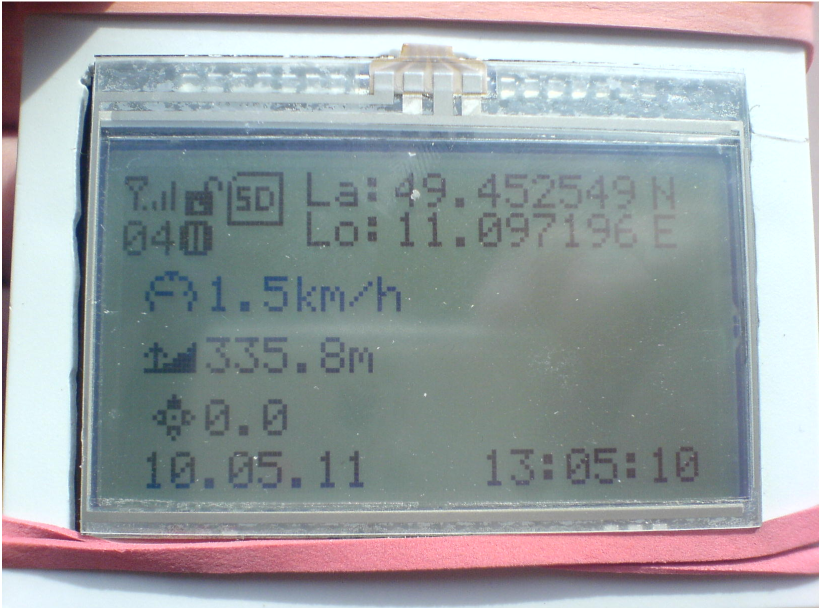
- dimmable backlight via PWM
- turning GPS receiver on/off via Mosfet
- one board instead of our 4 layer solution
- lipo batteries (flat)

Ideas for future development

Software:

- Integration of an OS (e.g. μ OS)
- Implementation of further touch screen interaction applications
- Rework to a 3-layer-architecture (Driver - Driver-Interfaces - Application)
- Navigation features (e.g. Waypoints, Statistics,...)
- PC application for communication via RS232
- automated generated API documentation (Doxygen)

Hardware receiving data...



Reference

- http://www.lcd-module.de/pdf/grafik/dogl128-6.pdf, Electronic Assembly
- NL_u-blox5_Referenzmanual_06102008_571.pdf, u-blox 5 NMEA, UBX Protocol Specification
- http://www.atmel.com/dyn/resources/prod_documents/doc2593.pdf ATmega644L datasheet