

Autonomous Environmental Buoy
and
Regatta Beacon

User Manual



Table of contents

1	Introduction.....	2
2	Safety first.....	3
3	System Specification.....	3
3.1	CTD Sensor.....	4
3.2	Wind Sensor.....	6
3.3	GNSS receiver.....	6
3.4	Communication module.....	7
3.5	Antennas.....	8
3.6	Interfaces.....	8
3.7	Power supply.....	9
3.8	Data storage.....	11
3.9	Mooring System.....	11
3.10	Physical implementation.....	12
3.11	Software.....	13
4	Operating Instructions.....	13
4.1	Installing/Mooring.....	13
4.2	Switch on/off.....	14
4.3	Receive measured data.....	14

1 Introduction

This is the user manual for the first prototype of the Autonomous Environmental Buoy/Regatta Beacon. The buoy's function is to measure, store and send environmental data.

It can serve in a river, or lake, but not in the sea or ocean. The first prototype consists of:

- fibreglass hull (Figure 1-1),
- fibreglass tier,
- stainless steel frame,
- STM32F3 Discovery MCU,
- ISEP-IPP CTD sensor,
- Davis Anemometer - wind speed and direction sensor,
- Parallax microSD card socket,
- Novatel SUPERSTAR II GNSS receiver,
- mooring system with Danforth anchor for sandy riverbeds.



Figure 1-1 The fibreglass hull

This prototype has the following features:

- calculated and tested buoyancy – it can carry up to 40kg
- waterproof improvements (rubber tape, fibreglass reinforcements for bolts, rubber washers for bolts),
- fibreglass tier to mount inner boxes with components,
- steel structure for attaching all outer components (like CTD, wind sensor),
- complete mooring system than can be connected to the steel structure,
- STM32F3 Discovery MCU with working built-in E-compass,
- Davis Anemometer for wind conditions measurement,
- high power efficiency; pre-calculated power consumption,
- developed powering system,
- developed interfaces connections system,
- designed inner layout of components,
- blinking LED lamp for visibility.

2 Safety first

In order to assure safety and not cause any damage, please:

- **DO NOT** touch any metal parts concerning electronics while the power supply is connected to the system - risk of electric shock;
- **DO NOT** move the buoy with less than two people;
- **DO NOT** place the buoy into water while electronic parts are inside - the prototype's hull is not completely waterproof;
- **DO NOT** expose the hull to the close contact with fire;
- **DO NOT** connect any components and parts without reading this user manual.

3 System Specification

The overall system concept is presented in Figure 3-1:

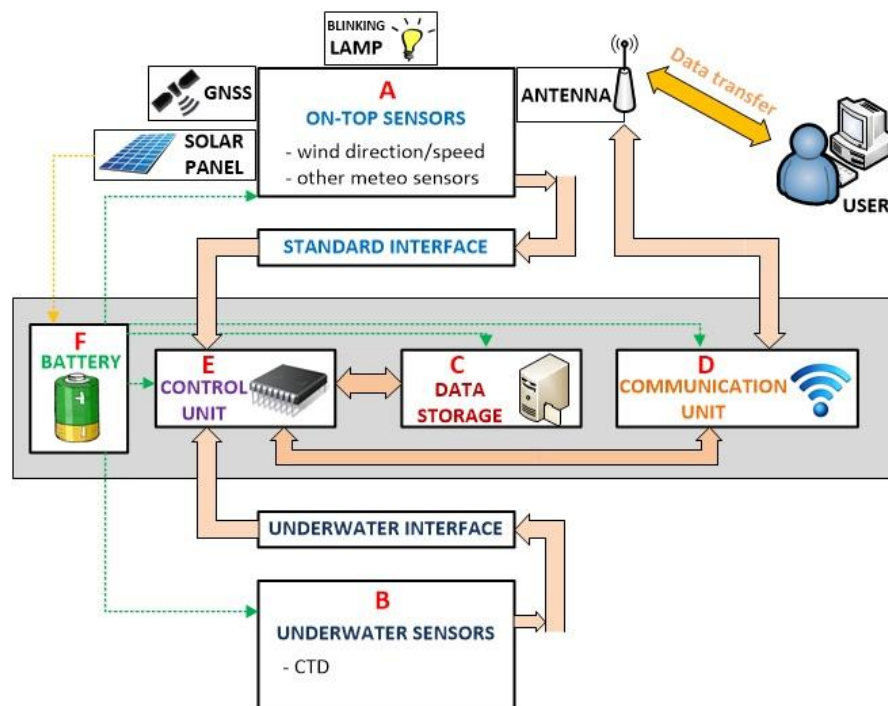


Figure 3-1 Buoy's overall system concept

Block A represents an upper steel platform, where sensors like Davis Anemometer, GNSS antenna or blinking lamp can be placed. There is also an option to install additional antenna for communication unit to extend the signal power. Solar panel is not available but it can be installed. Block B represents the second steel platform situated under the water surface where all underwater sensors can be attached. At this point, only the CTD sensor is available. The grey area in the figure above gathers all electronic parts which need to be placed inside the

fibreglass hull. In the first prototype these are: the MCU, batteries, the GNSS receiver, the microSD card socket.

The system concept is the following:

- sensors receive data and send them via standard interfaces to the control unit,
- MCU processes data and write them in readable format on the microSD card (Figure 3-2),
- on request, data is sent via control unit to the outer device ,
- meanwhile the battery provides power to all devices,
- blinking lamp blinks continuously.

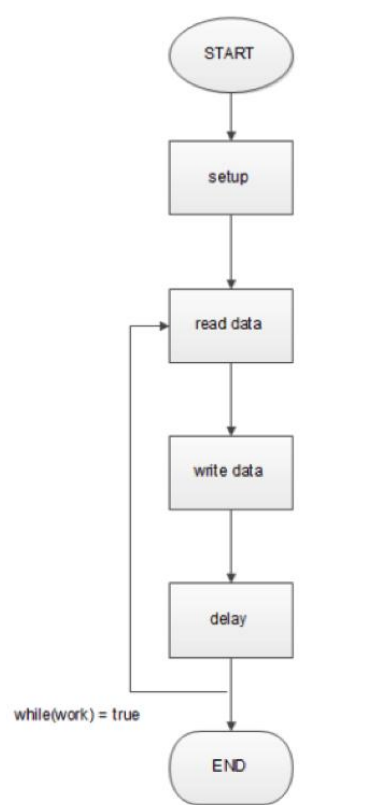


Figure 3-2 Flowchart of the data processing

The first prototype does not contain a communication unit. Only receiving data from wind direction sensor, E-compass and GNSS receiver was tested.

3.1 CTD Sensor

The CTD sensor may be placed in the lower part of the steel structure. It has an external thread that can go through the hole in the centre of the lower circular plate (which has to be

drilled), and can be then fastened with a spring washer and hexagon nut. The cable for the power supply and signals can then be connected to the bulkhead. Detailed specifications of the CTD sensor can be found in Table 3-1.

Table 3-1 CTD specifications

Product name	CTD		
Size	366 x Ø95 mm		
Power supply	8-30V		
Power consumption	1,5 W/12 V		
Communication	RS232 or I2C		
Quantity	Conductivity	Temperature	Pressure
Sensor	7 electrodes	PT 10	Oad cell
Unit	mS/cm	°C	dbar
Range	0-70 mS/cm	-5 °C to 35 °C	0-100 dbar
Resolution	+/- 0:01 mS/cm	+/- 0,001 °C	0,2% full scale

Communication:

The system communicates with the PC or other control unit via serial port. The data is sent at a transmission rate of 9600 b/s in ASCII format. The communication of the CTD is full-duplex and to send data the system requires the id card. This information is obtained by sending the command "\$? <cr>". The CTD allows sending data continuously or only when requested. In addition, it is possible to receive the data already processed (the salinity) or "raw data" for further treatment. There are many other features such as changing the id card, changing the transmission rate, downloading the calibration coefficients for the "raw data", etc. In continuous mode, data are sent at intervals of 100 ms using the following format:

!id#+ppp.ppp,+tt.ttt,+cc.ccc,+ss.ss<cr> ,

where:

- **id** (2-3) - Id of the card,
- **ppp.ppp** (5-12): Value of absolute pressure (dbar),
- **tt.ttt** (14-20): Value of temperature (°C),
- **cc.ccc** (22-28): Value of conductivity (mS/cm),
- **ss.sss** (30-35): Value of salinity,
- **<cr>**: Carriage Return (hex 0D).

All information regarding the CTD sensor can be found in [1].

3.2 Wind Sensor

The wind sensor may be attached with four bolts and nuts to the L-shaped vertical bar. The L-shaped vertical bar can be in turn attached to the upper circular plate with two bolts. The specifications of the wind sensor can be found in Table 3-2 below.

Table 3-2 Wind sensor specifications

Product name	Davis anemometer	
Dimensions	470 mm x 191 mm x 121 mm	
Weight	1,332 kg	
Connector	Modular connector (RJ-11)	
Cable length	12 m	
Cable type	4-conductor, 26 AWG	
Measurement of	Wind speed	Wind direction
Sensor type	Solid state magnetic sensor	Wind vane and potentiometer
Accuracy	+/- 3 km/h	+/- 7°
Resolution	0,1 m/s	1° (0° to 355°)
Range	1 to 322 km/h	0° to 360°
Sample period	2,25 seconds	1 second
Output	1600 rev/hr = 1 mph $V = P(2,25/T)$ V – speed in mph P - # of pulses per sample period T – sample period in seconds	Variable resistance 0 – 20 kΩ; 10 kΩ = south, 180°

All information regarding Davis anemometer can be found in [2]

3.3 GNSS receiver

The GNSS receiver used in this buoy is the NovAtel Superstar II. Its specifications are listed in Table 3-3

Table 3-3 GNSS receiver

Product name	Superstar II		
Size	46 x 71 x 13 mm		
Weight	22 g		
Input voltage	+3,3 or + 5 VDC		
Power consumption	0,8 W		
Communication ports	1 TTL serial port capable of 300 to 19200 bps		
	1 TTL DGPS port capable of 300 to 19200 bps		
Input/output connector	20-pin dual-row male header		
Antenna input	MCX female		
Operating temperature	-30 °C to 75 °C		
Humidity	5% to 95%		
Signal reacquisition	< 1s		
Position accuracy	Single point L1	WAAS L1	DGPS (L1, C/A)
	< 5m CEP	< 1,5 CEP	<1m CEP
Measurement precision	L1 C/A code		L1 carrier chase
	75 cm RMS		1 cm RMS
Data rate	Measurements		Position
	5 Hz		5 Hz

Superstar II is able to output data in NMEA format.

All information regarding NovAtel Superstar II receiver can be found in [3] and in [4].

3.4 Communication module

This prototype does not contain a communication module. Yet, to perform some calculations, the RN-XV WiFly Module with RPSMA connector was considered (it can be found at www.inmotion.pt). The important remark to make is that, in order to assure the best performance and power efficiency near water surface, 5 GHz Wi-Fi modules (instead of 2.4 GHz) should be considered.

3.5 Antennas

To enhance the performance of GNSS receiver, a GNSS antenna placed on the upper steel platform is required. Active antennas, such as Novatel GPS-701-GG, are recommended.

To extend the Wi-Fi communication signal power an additional RF antenna may be installed on the upper platform. This solution is optional. The use of an additional antenna depends on performance of the Wi-Fi module put inside the fibreglass hull.

3.6 Interfaces

To provide the possibility of attaching new sensors, the following interfaces are available:

- 3 USARTs (which may be used for RS232 interfaces),
- 1 SPI,
- 1 CAN,
- 1 USB,
- 1 RJ-11 (which consists of 2 signal lines: digital and analogue).

Detailed interface system is presented in Figure 3-3.

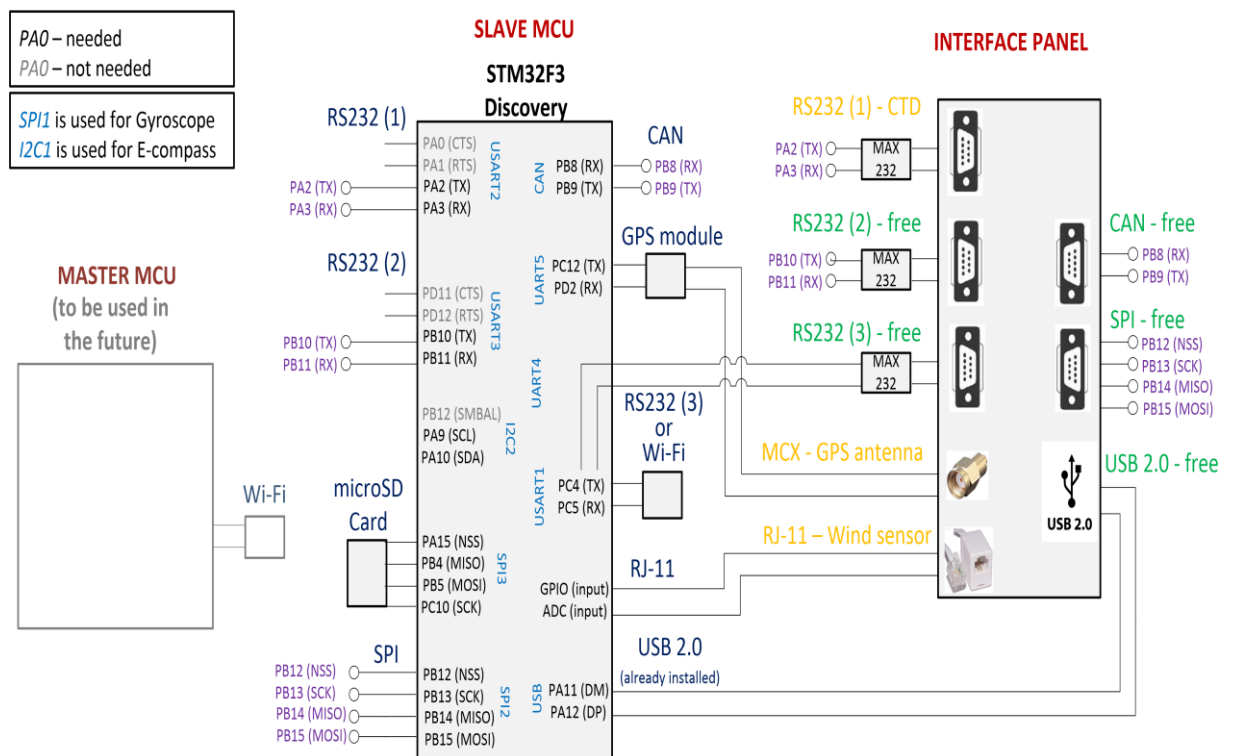


Figure 3-3 Signal schematic

All interfaces are provided by STM32F3 Discovery board. One of SPIs is used for built-in E-compass, the second one is dedicated for microSD card socket, the third one remains free. 3 USARTs are intended to be used for RS232 interfaces, yet, while only one MCU is considered, one USART has to be dedicated for connecting the Wi-Fi module. One of RS232 interfaces is occupied by CTD sensor. Fourth USART has to be used for GNSS receiver connection. Fifth USART cannot be utilized at all, while all three SPIs are in use. RJ-11 interface can be realized by one ADC available on the MCU board and one of GPIOs. More information regarding STM32F3 Discovery board interfaces can be found in [5].

An interface panel, seen on the right side of the Figure 3-3, has not been implemented yet physically. It can be realised by attaching waterproof connectors, such as Bulign [6], on the side of the fibreglass hull.

3.7 Power supply

Devices considered in the prototype need the following DC voltage levels: 12 V, 5 V, and 3.3 V (Table 3-4).

Table 3-4 Operating voltage of devices

Device	Voltage needed	Voltage regulator used
CTD	12 V	None
Blinking lamp		
STM32F3 Discovery	5 V	6 V to 5 V linear regulator
2nd MCU (optional) *		
MAX232 converters *		
GNSS antenna (MCX connector) *		
Wi-Fi module *	3.3 V	6 V to 3.3 V linear regulator
microSD card socket		
GNSS receiver		
Wind sensor (RJ-11 connector)		

* component not included in the prototype

Recommended power source is the 6 V NiMH battery. 5 V and 3.3 V levels may be provided by linear voltage regulators: 6 V to 5 V (L7805CV) and 6 V to 3.3 V (LM1086). 12 V level may be realised by serial connection of two 6 V batteries. To enable installing the RS232 interfaces, TTL to RS232 voltage level converters are needed (for instance MAX232 ICs). The ideological schematic of power connections is presented in Figure 3-4.

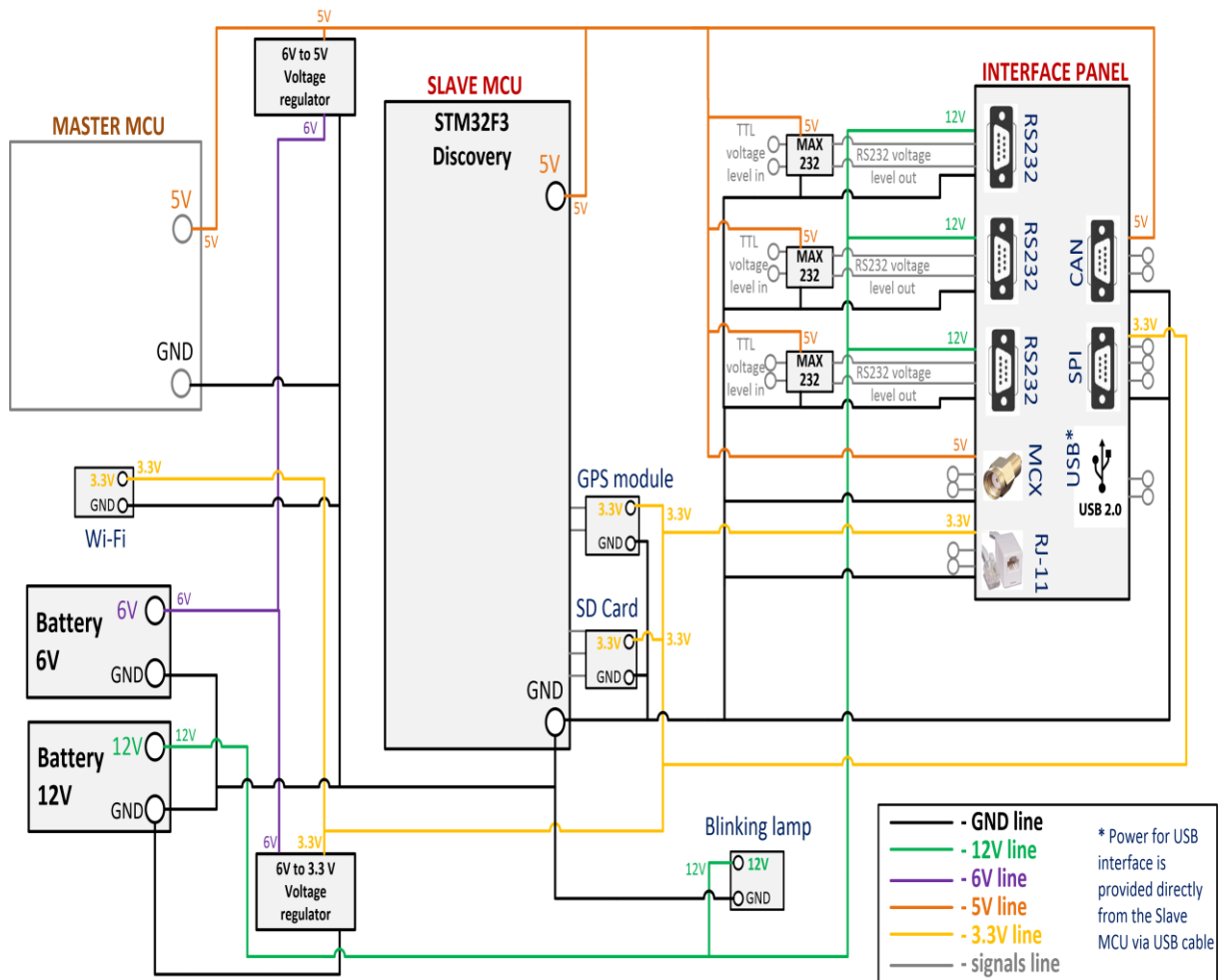


Figure 3-4 Power schematic of the buoy

Master MCU seen on the left side of the schematic is not included in this prototype.

All components has to be connected to the common ground of the STM32F3 Discovery board.

The power consumption of this system, assuming continuous work of all devices and taking into consideration power dissipation in voltage regulators, is around 6.3 W. The current load of 6 V to 5 V regulator is in this case 63.536 mA and 6 V to 3.3 V regulator is 240.697 mA.

3.8 Data storage

The best way to store data in this system is to use microSD card. The Parallax microSD card socket can be connected to the MCU via SPI. To ensure easy-operational system, use of FAT file system on the SD card is recommended.

3.9 Mooring System

The mooring system consists of a total of 10 parts, as seen in Figure 3-5. Starting from the lowest, these are:

- the 10 kg anchor,
- the swivel,
- the 3,5 m x 6 mm chain,
- the shackle INOX 10 mm,
- the 50 m x 12 mm nylon rope,
- two stainless steel rope clamps INOX 10 mm,
- the thimble INOX 10 mm,
- the swivel INOX 10 mm,
- the eye bolt INOX 12 x 120 mm.



Figure 3-5 Mooring system configuration

3.10 Physical implementation

All electronic devices, except the CTD, blinking lamp, wind sensor and antennas, should be placed inside the white fibreglass hull. To make sure they will be protected against water, they should be put inside a small waterproof box such as GEWISS watertight box [7]. On the top of the box two wholes should be drilled: one for power wiring, second for signal wiring. Wires may be put together into a bigger cable which protects them against water. One of these cables should go from an electronic box into a battery box and should contain at least three wires: 3.3 V line, 5 V line, GND line. To make the battery box easily removable form the hull, the cable has to be connected via waterproof connector, such as Bulgin [6]. The second of these cables should go from a battery box up to the interface panel on the upper side of the hull and should contain at least four wires: 3.3 V line, 5 V line, 12 V line, GND line. It also has to be easily removable so the use of Bulgin connector is necessary. The last (third) cable should be linked between electronics box and interface panel. This one does not have to be easily removable (so use of special connector is not needed) but it has to be thick enough to collect at least 15 wires sending signals to all interfaces.

The idea of having an interface panel on the hull is to make the buoy more reconfigurable. The panel should have all interfaces connectors (all of them waterproof). If any new sensor is needed to attach, the white hull does not have to be opened. The layout of this system is simply presented in Figure 3-6.

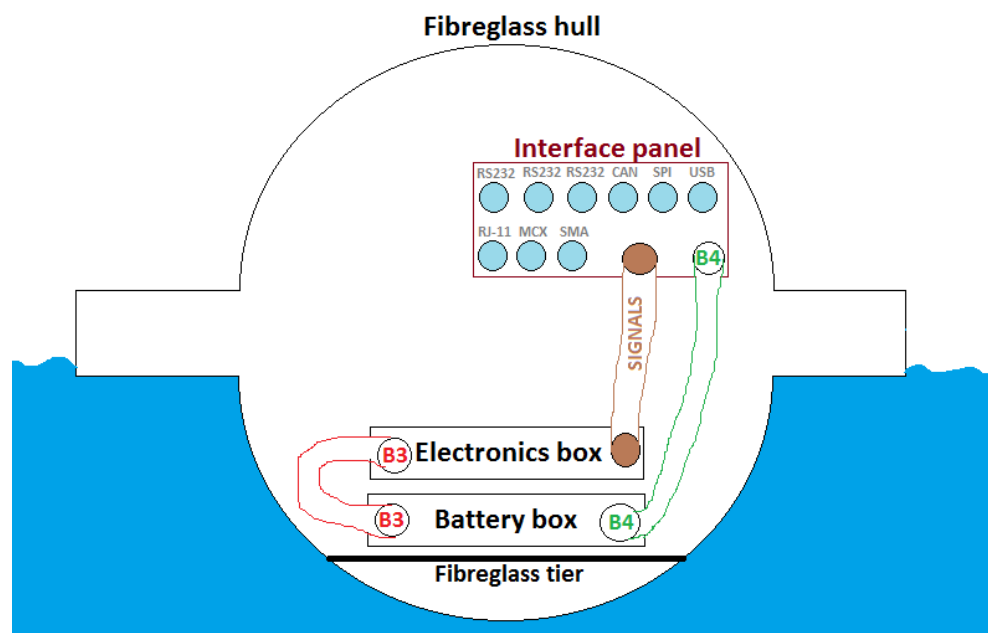


Figure 3-6 Inner layout of the components

B3 is a Bulgin 3-pin connector, **B4** is a Bulgin 4-pin connector. **Red**, **brown** and **green** lines represent thick cables gathering small wires. In order to change the batteries, the electronics box can be moved to the side, all B3 and B4 connectors can be disconnected and the whole battery box can be taken out of the hull. To make it possible, all wires and cables have to be long enough.

Because the battery box will be the heaviest component inside the hull, it has to be placed under the second box. As far as stability is concerned, the lower the heaviest part is placed the better. Moreover, it should be placed as near to the centre as possible. Size of both boxes makes it impossible to put them next to each other.

3.11 Software

For communicating with the MCU the IAR Embedded Workbench application using C language is recommended. All necessary libraries and some exemplary programs for STM32F3 Discovery can be downloaded from:

<http://www.st.com/web/en/catalog/tools/PF258154>

4 Operating Instructions

4.1 Installing/Mooring

The overall structure consists of four major parts: the fibreglass hull, the upper part of the steel structure, the lower part of the steel structure, and the mooring system. In order to have a functional buoy, all these parts need to be assembled in a correct manner. Even though the order in which they are put together is not important, for the sake of convenience we suggest that you follow these instructions. Firstly, deal with the mooring system. Place all components, as listed in section 3.9, close to each other. Take the bigger swivel and connect one of its end to the hole in the anchor. Then, attach the chain to the other end of the swivel. Afterwards, take the other end of the chain and connect it to the rope by applying the shackle

through the rope's thimble. Next, place the other end of the rope over the thimble so that it extends around 30 cm, apply one of the rope clamps close to the thimble, and the other 15 cm further. Lastly, take the second swivel and pass it through the thimble and the eye bolt. Once the mooring system is finished, move on to the steel structure and the hull. Lay down the hull and both parts of the steel structure close to each other horizontally on the floor. Pick up the hull and place it in the lower part of the steel structure, and then match the upper part. Quickly place three bolts in the holes, apply three spring washers and hexagon nuts and fasten them. Having done all of this, screw in the eye bolt into the threaded hole in the lower part of the steel structure. Do it so that it is possible to screw in the "blocker" – the small screw that goes through the little hole in the cylindrical wall of the threaded hole and enter the groove in the eye bolt's thread.

4.2 Switch on/off

Due to the fact that the electronic system is not complete, there is no switch to turn on and off the system. Particular devices can be turn on and off by simply connecting and disconnecting the power supply, such as the battery or the PC.

4.3 Receive measured data

To receive data from sensors, the following steps have to be taken:

- connect the sensor to one of interfaces described in section 3.6,
- create a project in IAR Embedded Workbench,
- write a routine which initializes the desired interface,
- write a program which receives data from the interface,
- connect all devices to the power supply,
- display received data on the hyper terminal or write it to the microSD card.

For instance, a procedure of receiving data from wind direction sensor and displaying it on the computer screen is presented below.

Connect the Davis anemometer's wind direction sensor to the STM32F3 Discovery

According to Davis anemometer's datasheet [8], the green wire gives analogue DC voltage output. It has to be connected to one of MCUs ADCs, for instance to ADC1 which uses PC1 pin (Figure 4-1).

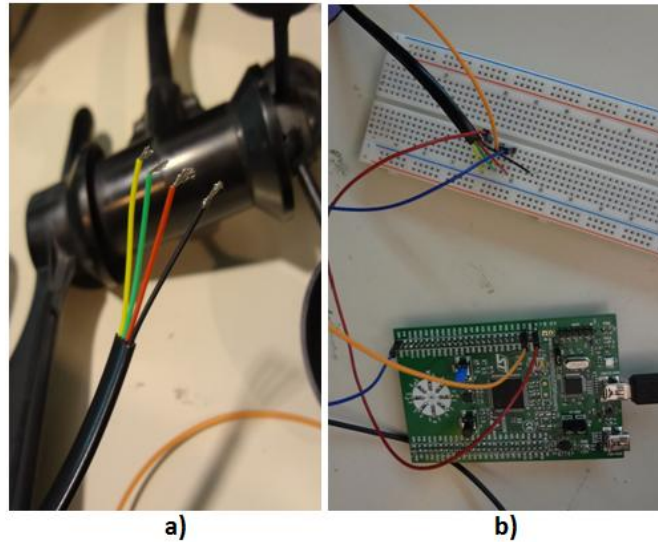


Figure 4-1 Wiring of the Anemometer:
a) sensor's wires; b) connection with MCU

Create a project in IAR Embedded Workbench

The process of creating and configuring new project is described in [9]. In this case the firmware example of ADC program can be used [10].

Write a routine which initializes the interface

In this case the ADC has to be initialized. The code for this routine is presented in Figure 4-2.

Write a program which receives data from the interface

In this case program, presented in Figure 4-3, compute received signal into voltage in mV and saves it in the variable *ADC1ConvertedVoltage*.

Connect all devices to the power supply

The STM32F3 Discovery board is powered by the connection with the PC via USB port. Davis anemometer is connected as follows: red wire to the F3 Discovery common ground, yellow wire to the F3 Discovery external 3V voltage pin (Figure 4-1 b).

<pre> /* long initialization procedure for ADC */ /* Configure the ADC clock */ RCC_ADCLKConfig(RCC_ADC12PLLCLK_Div2); /* Enable ADC1 clock */ RCC_AHBPeriphClockCmd(RCC_AHBPeriph_ADC12, ENABLE); /* Setup SysTick Timer for 1 µsec interrupts */ if (SysTick_Config(SystemCoreClock / 1000000)) { /* Capture error */ while (1) {} } /* ADC Channel configuration */ /* GPIOC Periph clock enable */ RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOC, ENABLE); /* Configure ADC Channel7 as analog input */ GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1 ; GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AN; GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL ; GPIO_Init(GPIOC, &GPIO_InitStructure); ADC_StructInit(&ADC_InitStructure); </pre>	<pre> /* Insert delay equal to 10 µs */ Delay(10); ADC_SelectCalibrationMode(ADC1, ADC_CalibrationMode_Single); ADC_StartCalibration(ADC1); while(ADC_GetCalibrationStatus(ADC1) != RESET); calibration_value = ADC_GetCalibrationValue(ADC1); ADC_CommonInitStructure.ADC_Mode = ADC_Mode_Independent; ADC_CommonInitStructure.ADC_Clock = ADC_Clock_AsynClkMode; ADC_CommonInitStructure.ADC_DMAAccessMode = ADC_DMAAccessMode_Disabled; ADC_CommonInitStructure.ADC_DMAMode = ADC_DMAMode_OneShot; ADC_CommonInitStructure.ADC_TwoSamplingDelay = 0; ADC_CommonInit(ADC1, &ADC_CommonInitStructure); ADC_InitStructure.ADC_ContinuousConvMode = ADC_ContinuousConvMode_Enable; ADC_InitStructure.ADC_Resolution = ADC_Resolution_12b; ADC_InitStructure.ADC_ExternalTrigConvEvent = ADC_ExternalTrigConvEvent_0; ADC_InitStructure.ADC_ExternalTrigEventEdge = ADC_ExternalTrigEventEdge_None; ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right; ADC_InitStructure.ADC_OverrunMode = ADC_OverrunMode_Disable; ADC_InitStructure.ADC_AutoInjMode = ADC_AutoInjec_Disable; ADC_InitStructure.ADC_NbrOfRegChannel = 1; ADC_Init(ADC1, &ADC_InitStructure); /* ADC1 regular channel7 configuration */ ADC_RegularChannelConfig(ADC1, ADC_Channel_7, 1, ADC_SampleTime_7Cycles5); /* Enable ADC1 */ ADC_Cmd(ADC1, ENABLE); /* wait for ADRDY */ while(!ADC_GetFlagStatus(ADC1, ADC_FLAG_RDY)); /* Start ADC1 Software Conversion */ ADC_StartConversion(ADC1); </pre>
a)	b)

Figure 4-2 ADC initialization routine
a) first part; b) second part

```

/* Test EOC flag */
while(ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC) == RESET);

/* Get ADC1 converted data */
ADC1ConvertedValue =ADC_GetConversionValue(ADC1);

/* Compute the voltage */
ADC1ConvertedVoltage = (ADC1ConvertedValue *3300)/0xFFF;

```

Figure 4-3 Voltage computing example

Display received data on the hyper terminal

To do that the serial connection between MCU and PC is needed. It can be obtained by connecting MCU's USART interface to the PC via USART-->USB connector. For that purpose USART1, using PC4 pin as Tx line and PC5 pin as Rx line, can be configured. Needed initializing routine is shown in Figure 4-4.

```

void USART1_config(void) {

    GPIO_InitTypeDef GPIO_InitStructure;
    USART_InitTypeDef USART_InitStructure;

    /* USARTx configuration -----*/
    /* USARTx configured as follow: BaudRate = 9600 baud; Word Length = 8 Bits;
    one Stop Bit; No parity; Hardware flow control disabled
    RTS and CTS signals); Receive and transmit enabled */

    USART_InitStructure.USART_BaudRate = 9600;
    USART_InitStructure.USART_WordLength = USART_WordLength_8b;
    USART_InitStructure.USART_StopBits = USART_StopBits_1;
    USART_InitStructure.USART_Parity = USART_Parity_No;
    USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
    USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;

    /* Enable USART clock */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1, ENABLE); // for USART1

    /*Enable or disable the AHB peripheral clock */
    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOC, ENABLE); // for USART1

    /*Configure GPIO pin alternate function */
    GPIO_PinAFConfig(GPIOC, GPIO_PinSource4, GPIO_AF_7);

    /*Configure GPIO pin alternate function */
    GPIO_PinAFConfig(GPIOC, GPIO_PinSource5, GPIO_AF_7);

    /** USART1 GPIO Configuration
    /*Configure GPIO pin PC4 -----> USART1_TX */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOC, &GPIO_InitStructure);

    /*Configure GPIO pin PCS -----> USART1_RX */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;
    GPIO_Init(GPIOC, &GPIO_InitStructure);

    /* USART configuration */
    USART_Init(USART1, &USART_InitStructure); // for USART1

    /* Enable USART */
    USART_Cmd(USART1, ENABLE); // for USART1
}

```

Figure 4-4 Initializing USART1

The main body of the program which receives data from ADC and sends it to the hyper terminal via USART interface each 1 second may look as presented in Figure 4-5.

To display data received from USART -->USB connector any type of hyper terminal software can be used, for example Tera Term Pro.

```

int main(void)
{
    /* pinout configuration and initialization of USART1*/
    USART1_config();

    /* variables needed to print on hyper terminal */
    char buffer_Tx[20];
    char p_buffer;
    char *value = "text and number 2 and i can add even more";
    u8 RxData;

    while (1)
    {
        /* Test EOC flag */
        while(ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC) == RESET);

        /* Get ADC1 converted data */
        ADC1ConvertedValue =ADC_GetConversionValue(ADC1);

        /* Compute the voltage */
        ADC1ConvertedVoltage = (ADC1ConvertedValue *3300)/0xFFF;

        sprintf (buffer_Tx, "\t voltage: %d mV", ADC1ConvertedVoltage);
        p_buffer = 0;

        while (buffer_Tx[p_buffer] != 0)
        {
            // print on serial monitor using USART1
            while (USART_GetFlagStatus(USART1, USART_FLAG_TXE) == RESET)
            {}
            USART_SendData(USART1, buffer_Tx[p_buffer]);
            p_buffer++;
        }
        // 1 second delay
        Delay (1000000);
    }
}

```

Figure 4-5 Main body of an exemplary program

References

- [1] P. M. S. Guimarães, “Rede de Monitorização Ambiental em Tempo-Real com Sistemas Móveis,” 2009.
- [2] Davis, “Davis Anemometer documentation,” 2013. [Online]. Available: http://www.davisnet.com/weather/products/wx_product_docs.asp?pnum=07911. [Accessed 10 June 2013].
- [3] Novatel, “Novatel SUPERSTAR II datasheet,” 2006. [Online]. Available: <http://www.novatel.com/assets/Documents/Papers/SUPERSTAR.pdf>. [Accessed 10 June 2013].
- [4] Novatel, “Novatel SUPERSTAR II user manual,” 2005. [Online]. Available: <http://www.novatel.com/assets/Documents/Manuals/om-20000077.pdf>. [Accessed 06 June 2013].
- [5] STMicroelectronics, “User manual STM32F3DISCOVERY,” 2013. [Online]. Available: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00063382.pdf. [Accessed 2 April 2013].
- [6] Bulgin, “Bulgin connectors,” Bulgin, 2012. [Online]. Available: <http://bulgin.co.uk/>. [Accessed 03 June 2013].
- [7] GEWISS, “Surface-Mounting boxes GWPLAST 75,” 2010. [Online]. Available: <http://www.iasb.com.my/webshaper/store/viewProd.asp?pkProductItem=611>. [Accessed June 2013].
- [8] Davis, “Davis Anemometer datasheet,” 2013. [Online]. Available: http://www.davisnet.com/product_documents/weather/spec_sheets/7911_SS.pdf. [Accessed June 2013].
- [9] STMicroelectronics, “Getting started with STM32F3 Discovery kit,” 2012. [Online]. Available: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00062662.pdf. [Accessed June 2013].
- [10] STMicroelectronics, “STM32F3 Discovery kit firmware package,” 2013. [Online]. Available: <http://www.st.com/web/en/catalog/tools/PF258154>. [Accessed June 2013].

