

WATER CONNECT

water-analysis.git

TEAM



Amedeo Rosario Grandinetti
[LinkedIn: Amedeo Rosario Grandinetti](#)
[GitHub: AmedeoGrandi](#)
242507



Giorgio Ubbriaco
[LinkedIn: Giorgio Ubbriaco](#)
[GitHub: gubbriaco](#)
247284



Francesco Nicoletti
[LinkedIn: Francesco Nicoletti](#)
[GitHub: frembi99](#)
242530



Abstract

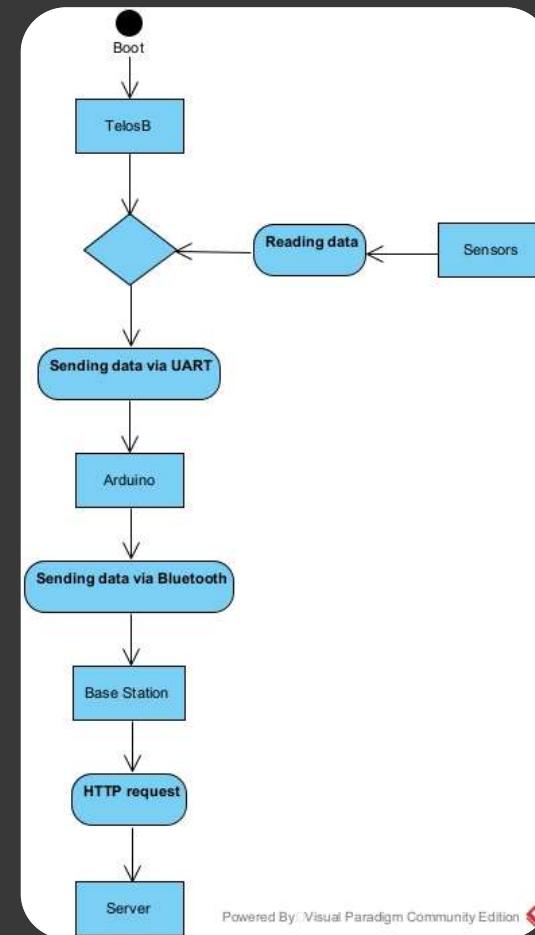
Contents

1. Design Concept
2. Hardware Design
3. Software Design
4. Prototype
5. Future Developments



Design Concept

1. TelosB booting
2. Reading data from sensors
3. Sending data from TelosB to Arduino via UART
4. Sending data from Arduino to Base Station via Bluetooth
5. Sending data from Base Station to Server via HTTP request



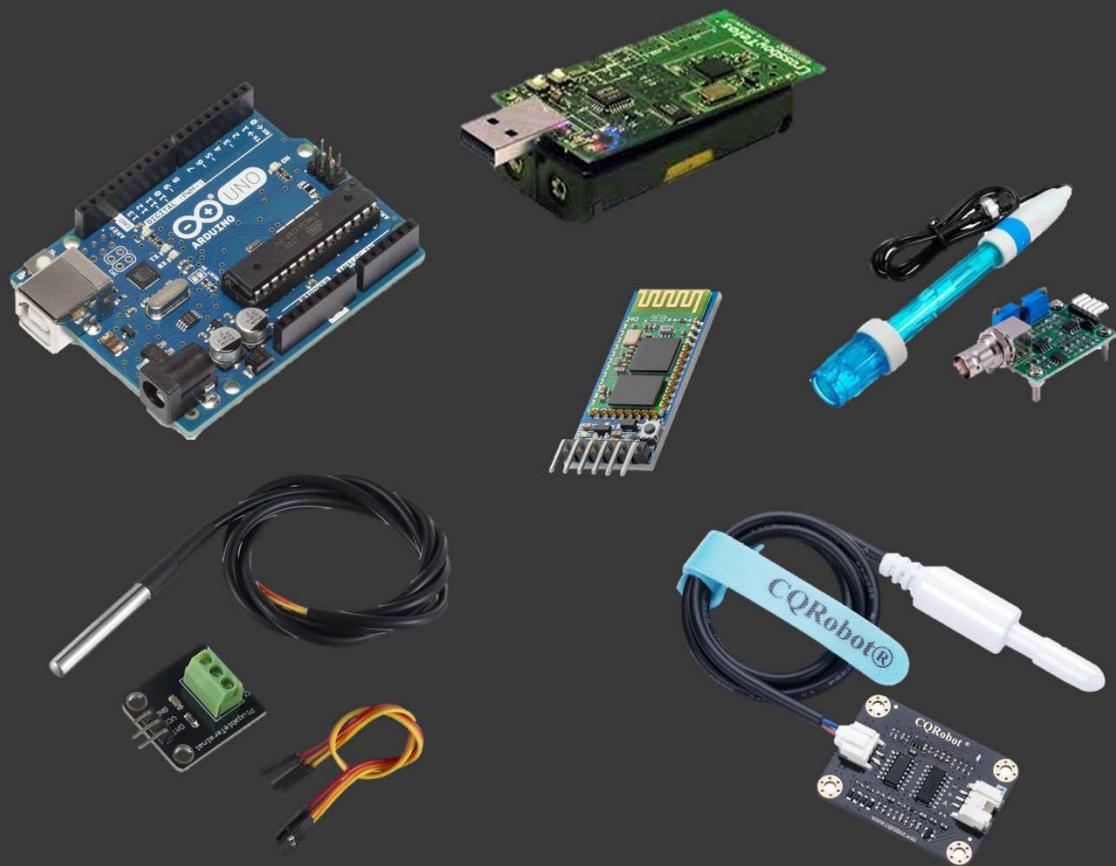


Design Choices

- Reading pH and TDS sensors with Telosb and Temperature sensor with Arduino
- Sending data from Telosb to Arduino through UART communication protocol
- Data processing with Arduino
- Sending processed data from Arduino to HC05 Bluetooth module
- Sending data from HC05 to a nearby basestation
- Sending data from basestation to an online server using HTTP requests
- Database building process with the data collection from three different geographic locations

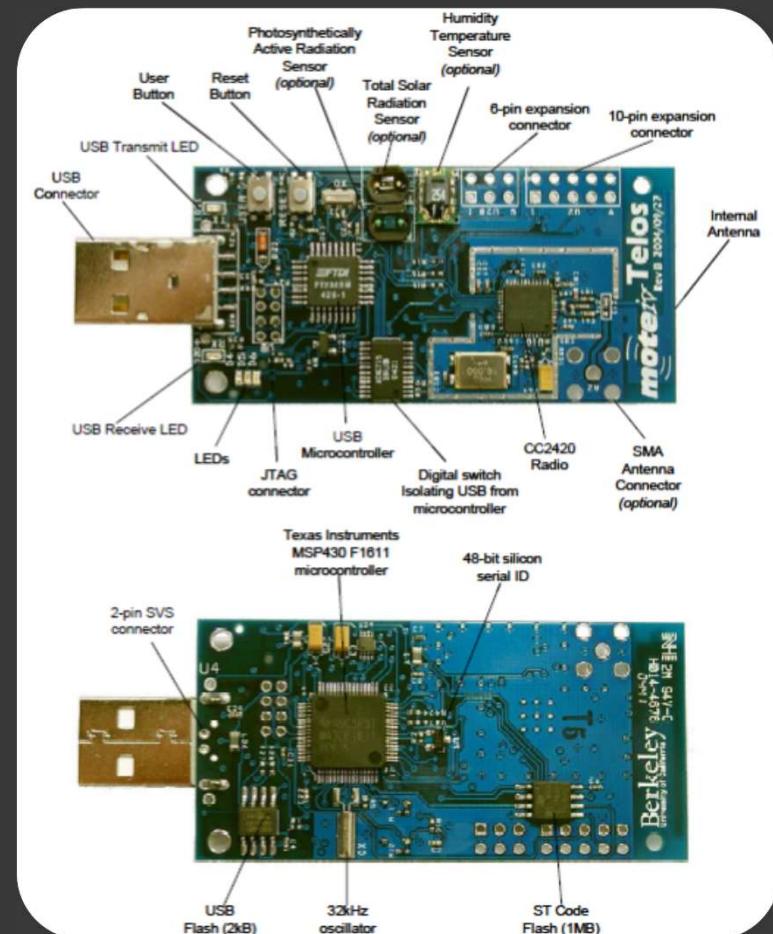
Hardware Design

1. TelosB
2. TDS CQRobot
3. pH E201 4502C
4. Arduino
5. DS18B20
6. HC-05



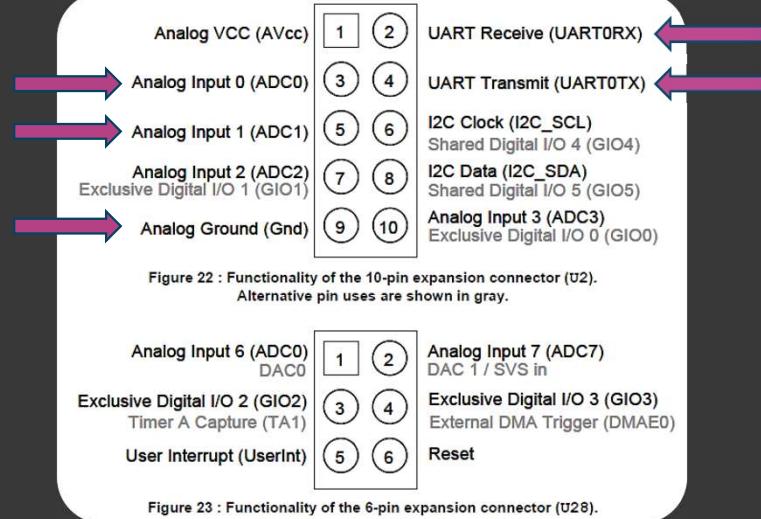
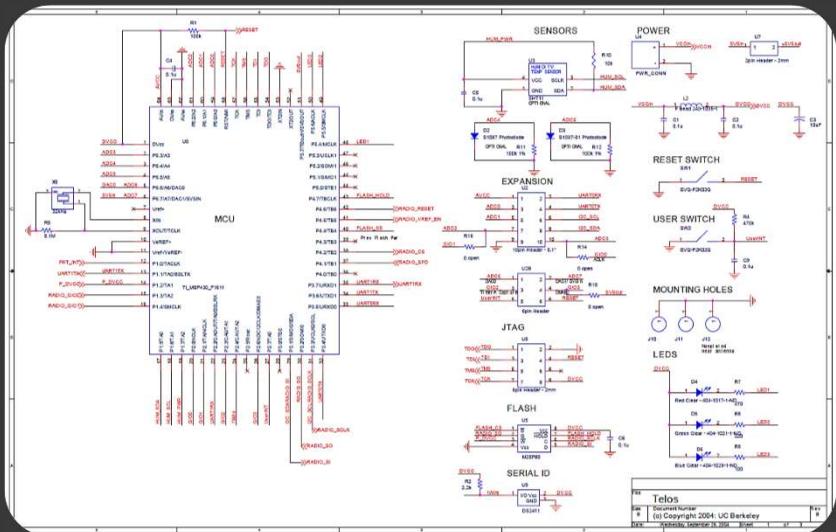
TelosB

- 250kbps 2.4GHz IEEE 802.15.4 Chipcon Wireless Transceiver
- Interoperability with other IEEE 802.15.4 devices
- 8MHz Texas Instruments MSP430 microcontroller
- Integrated ADC, DAC, Supply Voltage Supervisor, and DMA Controller
- Integrated Humidity, Temperature, and Light sensors
- Ultra low current consumption
- Programming and data collection via USB
- 16-pin expansion support
- TinyOS support : mesh networking and communication implementation



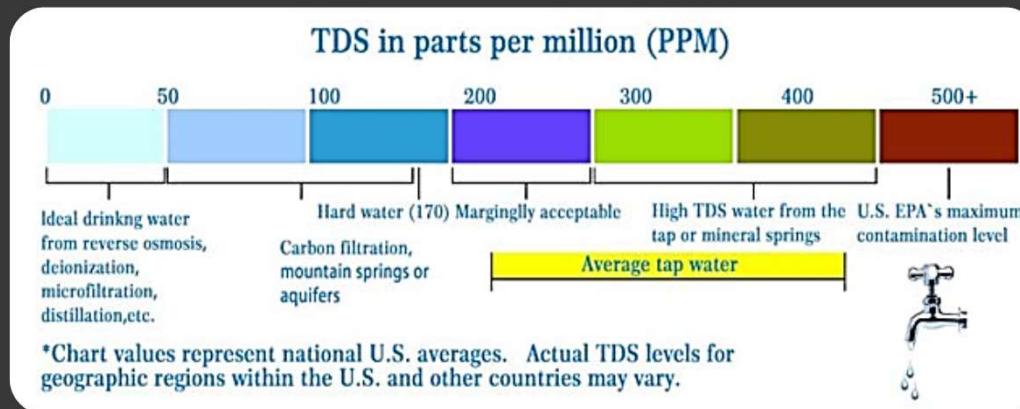
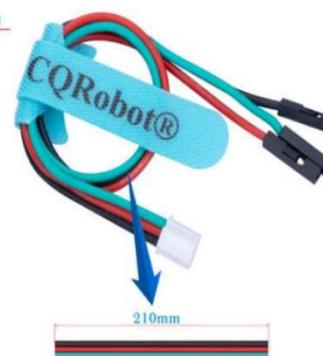
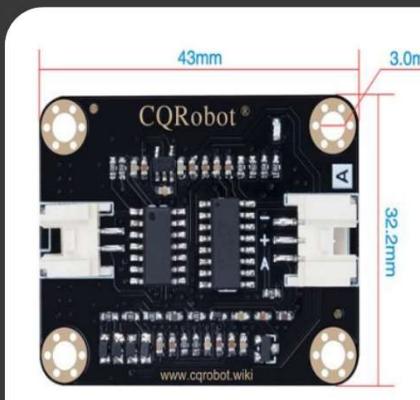
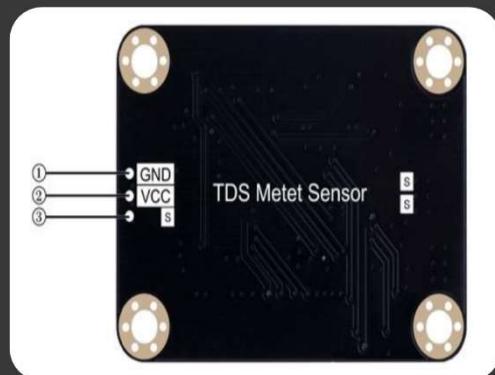
TelosB

- The 16-pin expansion connector can provide power to the module.
- 10-pin at position U2
- 6-pin at position U28
- 7 (U2) and 10 (U2) can be used as digital I/O if and only if R15 and R14 respectively are populated with a 0 ohm resistor.





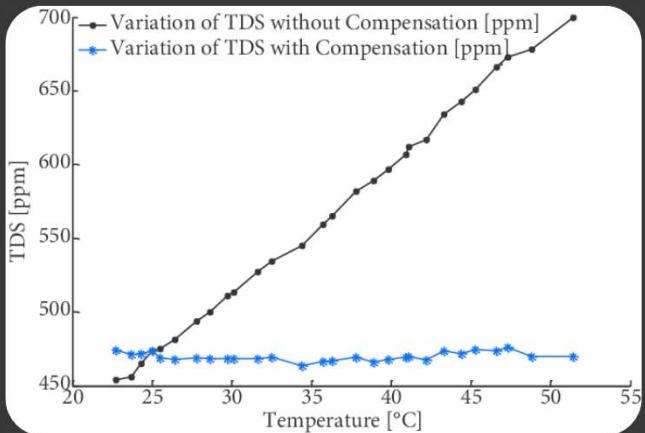
TDS CQRobot



- How much mg of solid substances are dissolved in one litre of water
- Returned values represented in particles per million (PPM)

https://www.cqrobot.com/index.php?route=product/product&product_id=1122

TDS CQRobot



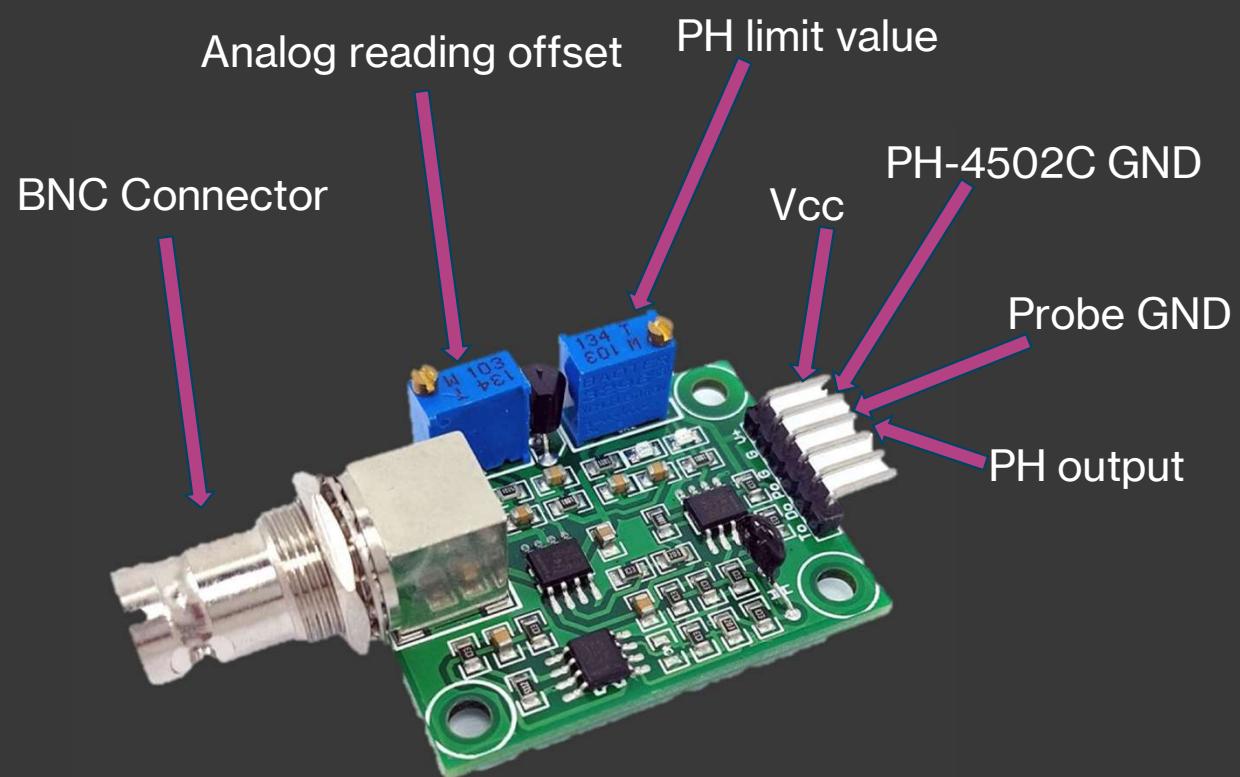
$$\text{compensationCoefficient} = 1.0 + 0.02 \cdot (\text{temperature} - 25.0)$$

$$\text{compensationVoltage} = \frac{\text{TDSaverageVoltage}}{\text{compensationCoefficient}}$$

$$\begin{aligned} \text{TDScompensated} \\ &= (133.42 \cdot \text{compensationVoltage}^3 - 255.86 \\ &\quad \cdot \text{compensationVoltage}^2 + 857.39 \\ &\quad \cdot \text{compensationVoltage}) \cdot 0.5 \end{aligned}$$



pH E201 4502C



<https://www.arduiner.com/it/prodotto/kit-modulo-sensore-ph-014-con-sonda-elettrodo-bnc-per-arduino/>

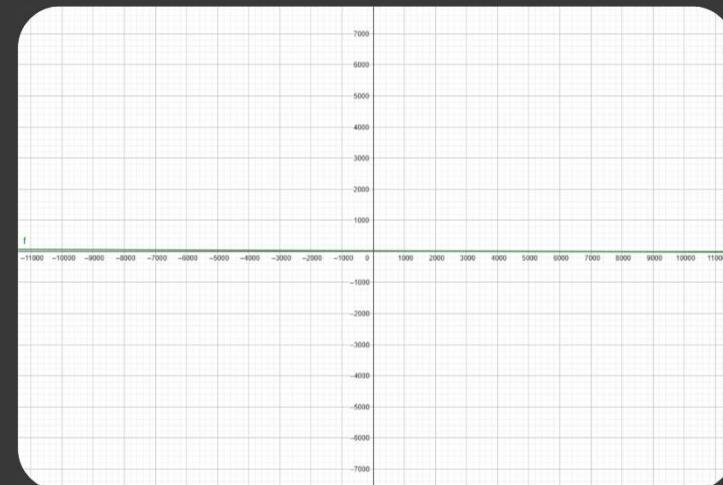
pH E201 4502C

$$\begin{aligned}
 slope &= \frac{pH(Water) - pH(Lemon)}{Voltage(Water) - Voltage(Lemon)} \\
 &= \frac{6.25 - 2.2}{2600 - 3645} = \frac{4.05}{-1045} = -0.003875598
 \end{aligned}$$

	pH	TelosB Voltage
Water	6.25	2600
Lemon	2.2	3645

$$\begin{aligned}
 calibration &= pH(Water) + [abs(slope) \cdot Voltage(Water)] \\
 &= 6.25 + (0.003875598 \cdot 2600) = 6.25 + 10.0765548 \\
 &= 16.3265548
 \end{aligned}$$

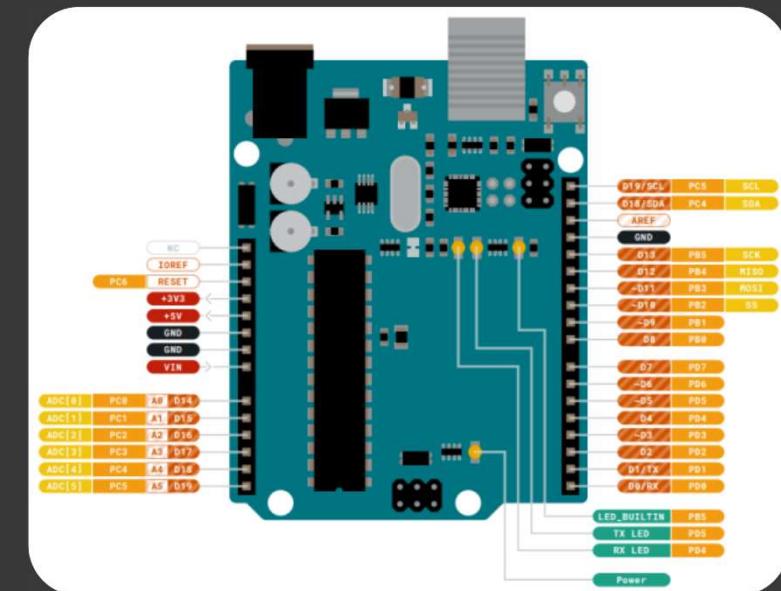
$$\begin{aligned}
 pH &= slope \cdot Voltage + calibration \\
 &= -0.003875598 \cdot Voltage + 16.3265548
 \end{aligned}$$



Arduino

Function	Type	Description
+3V3	Power	+3V3 Power Rail
→ +5V	Power	+5V Power Rail
→ GND	Power	Ground
→ GND	Power	Ground
A0	Analog/GPIO	Analog input 0 /GPIO
...	Analog/GPIO	..
A4/SDA	Analog input/I2C	Analog input 4 /I2C Data line
A5/SCL	Analog input/I2C	Analog input 4 /I2C Clock line
D0	Digital/GPIO	Digital pin 0 /GPIO
..	Digital/GPIO	..
→ D3	Digital/GPIO	Digital pin 3 /GPIO
D4	Digital/GPIO	Digital pin 4 /GPIO
→ D5	Digital/GPIO	Digital pin 5 /GPIO
→ D6	Digital/GPIO	Digital pin 6 /GPIO
→ D7	Digital/GPIO	Digital pin 6 /GPIO
→ D8	Digital/GPIO	Digital pin 6 /GPIO
→ D9	Digital/GPIO	Digital pin 6 /GPIO

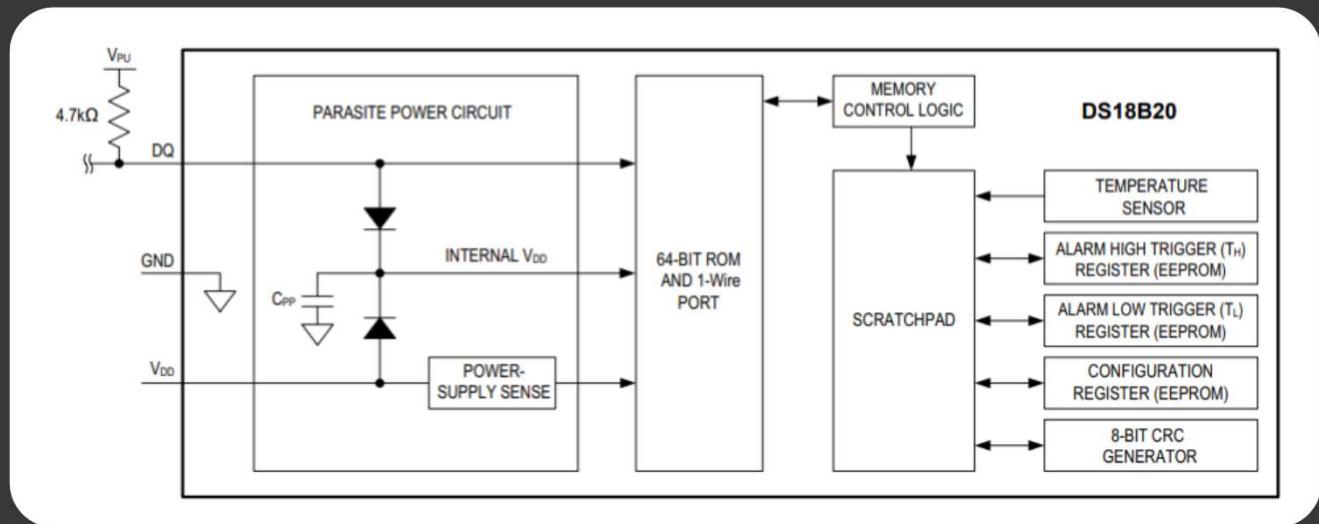
Function	Type	Description
D10	Digital input/SS	Digital pin 10 /SPI Chip Select
D11	Digital input/MOSI	Digital pin 11 /SPI Main Out Secondary In
D12	Digital input/MISO	Digital pin 12 /SPI Main Out Secondary Out
D13	Digital input/SCK	Digital pin 13 /SPI serial clock output



<https://store.arduino.cc/products/arduino-uno-rev3>



DS18B20





DS18B20

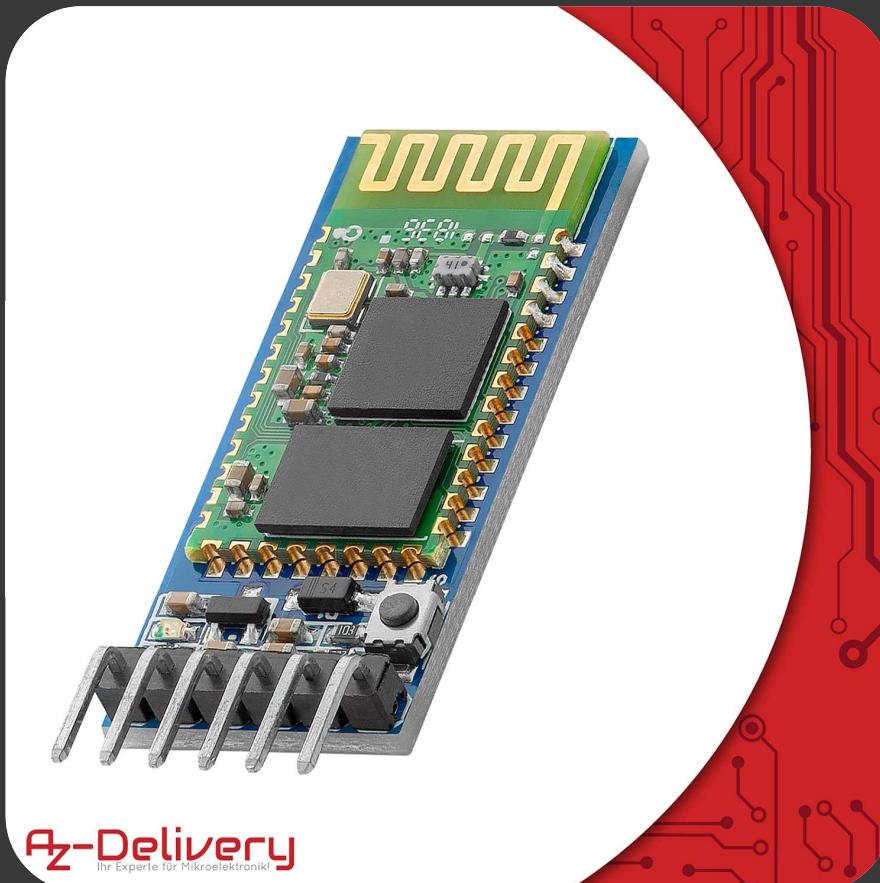
COMMAND	DESCRIPTION	PROTOCOL	1-Wire BUS ACTIVITY AFTER COMMAND IS ISSUED	NOTES
TEMPERATURE CONVERSION COMMANDS				
Convert T	Initiates temperature conversion.	44h	DS18B20 transmits conversion status to master (not applicable for parasite-powered DS18B20s).	1
MEMORY COMMANDS				
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	B Eh	DS18B20 transmits up to 9 data bytes to master.	2
Write Scratchpad	Writes data into scratchpad bytes 2, 3, and 4 (T_H , T_L , and configuration registers).	4 Eh	Master transmits 3 data bytes to DS18B20.	3
Copy Scratchpad	Copies T_H , T_L , and configuration register data from the scratchpad to EEPROM.	48h	None	1
Recall E ²	Recalls T_H , T_L , and configuration register data from EEPROM to the scratchpad.	B8h	DS18B20 transmits recall status to master.	
Read Power Supply	Signals DS18B20 power supply mode to the master.	B4h	DS18B20 transmits supply status to master.	

Used Libraries:

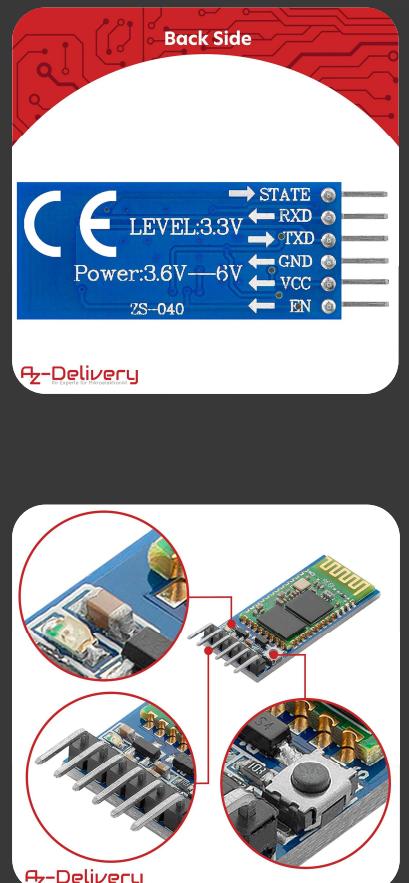
- [DallasTemperature.h](#)
- [OneWire.h](#)

HC-05

- Bluetooth SPP Module
- Low Power 1.8V Operation
- 1.8V to 3.6V I/O
- UART Interface with Programmable Baud Rate (Supported Baud Rate: 9600, 19200, 38400, 57600, 115200, 230400, 460800)
- Integrated Antenna
- Auto-Connect to the last device on power as default



Az-Delivery
Ihr Experte für Mikroelektronik



<https://www.az-delivery.de/it/products/hc-05-6-pin>



AT-COMMANDS

```
HC05setup.ino

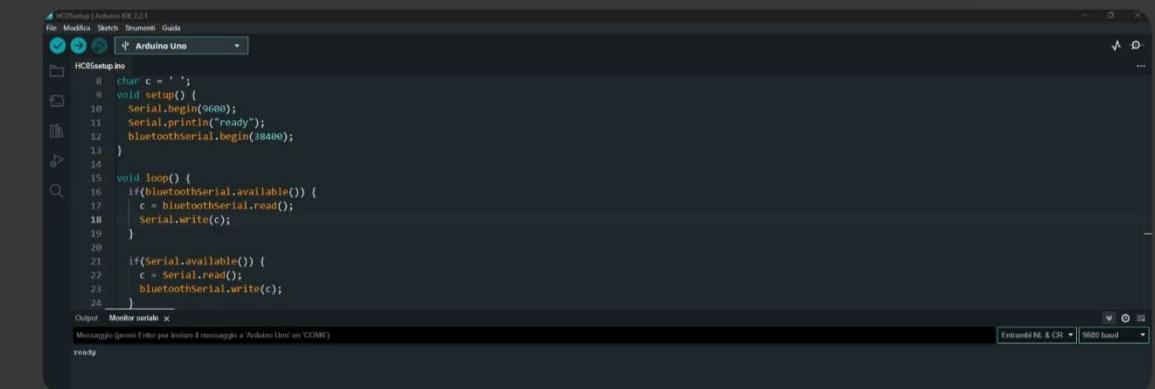
#include <SoftwareSerial.h>

#define BT_TX_PIN 11
#define BT_RX_PIN 10
SoftwareSerial bluetoothSerial(BT_RX_PIN, BT_TX_PIN);

char c = ' ';
void setup() {
    Serial.begin(9600);
    Serial.println("ready");
    bluetoothSerial.begin(38400);
}

void loop() {
    if(bluetoothSerial.available()) {
        c = bluetoothSerial.read();
        Serial.write(c);
    }

    if(Serial.available()) {
        c = Serial.read();
        bluetoothSerial.write(c);
    }
}
```



Put the Bluetooth module in **Setup (Configuration) Mode** and perform the following steps in the console:

1. AT
2. AT+ORGL
3. AT+RESET
4. AT+NAME=*device_name*
5. AT+ROLE=1
6. AT+PSWD='1234'
7. AT+UART=9600,0,0
8. AT+CMODE=0
9. AT+BIND=*base_station_bluetooth_address*



AT-COMMANDS

Function	AT Command	Response	Parameter	Observations	Examples
Test	AT	OK	None	-	AT OK
Restore default status	AT+ORGL	OK	None	The parameter of default status: 1. Device type: 0 2. Inquire code: 0x009e8b33 3. Module work mode: Slave Mode 4. Connection mode: Connect to the Bluetooth device specified 5. Serial parameter: Baud rate: 38400 bits/s; Stop bit: 1 bit; Parity bit: None. 6. Passkey: "1234" 7. Device name: "H-C-2010-06-01"	AT+ORGL OK
Set/Inquire device's name	AT+NAME=Param1	OK	Param1: Bluetooth device name	Length up to 32 bytes; Supports special characters; AT+NAME="HC-05" is the same as AT+NAME=HC-05 Default: "HC-05"	<i>Set the module device name to "HC-05":</i> AT+NAME=HC-05 r\n OK AT+NAME? r\n +NAME:HC-05 OK
	AT+NAME?	If Success: +NAME:Param1 OK If Failure: FAIL			
Set/Inquire module role	AT+ROLE=Param1	OK	Param1: module role: 0 -> Slave 1 -> Master 2 -> Slave-Loop	<i>Role introduction:</i> <i>Slave:</i> Passive connection; <i>Slave-Loop:</i> Passive connection, receive the remote Bluetooth master device data and send it back to the master device; <i>Master:</i> Inquire the near SPP Bluetooth slave device, build connection with it positively, and build up the transparent data transmission between master and slave device. Default: 0	<i>Set the module device role to Slave:</i> AT+ROLE=0 OK AT+ROLE? +ROLE=0 OK
	AT+ROLE?	+ROLE:Param1 OK			
Set/Inquire passkey	AT+PSWD=Param1	OK	Param1: Passkey	Default: "1234"	<i>Set the module device passkey to "1234":</i> AT+PWD=1234 r\n (or AT+PSWD="1234" r\n) OK AT+PSWD? r\n +PSWD:1234 OK
	AT+PSWD?	+PSWD:Param1 OK			



AT-COMMANDS

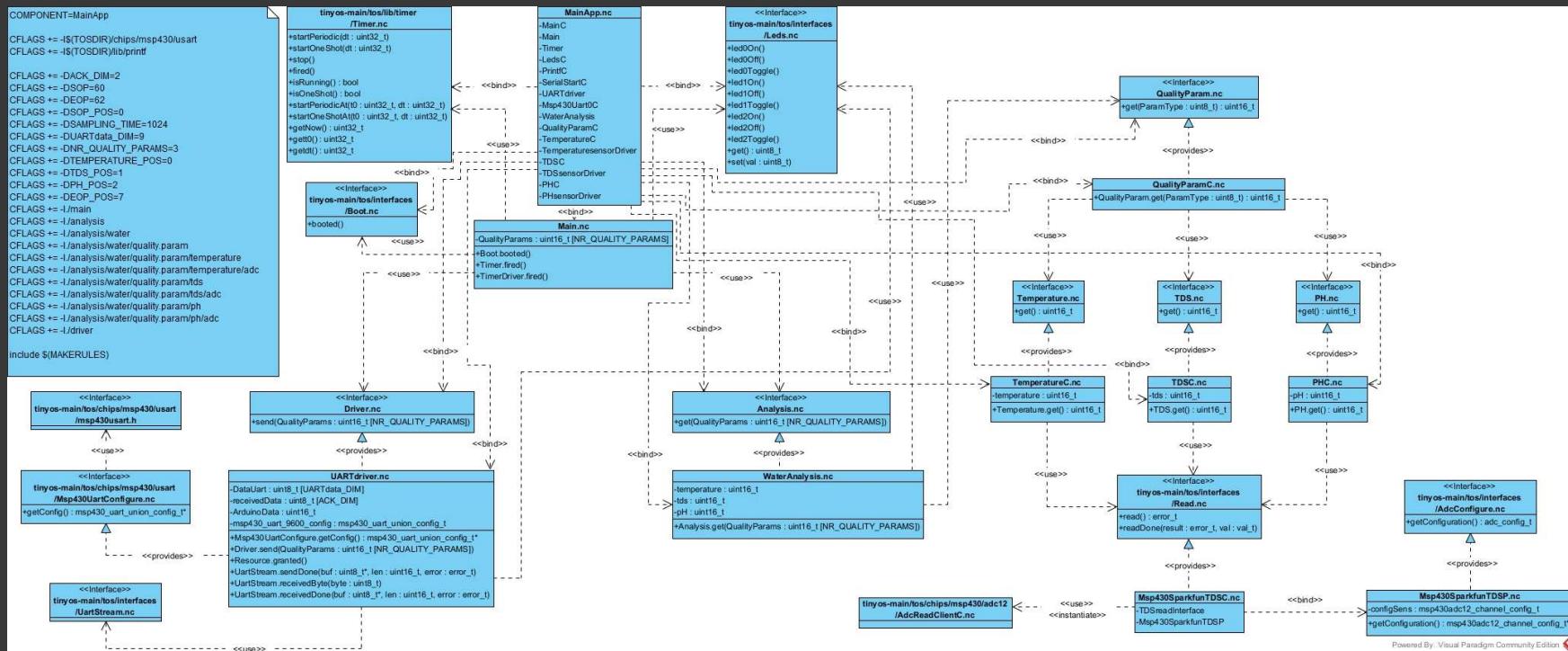
Function	AT Command	Response	Parameter	Observations	Examples
Set/Inquire serial parameter	7	OK	Param1: Baud rate: 4800 -> 4800 bits/s 9600 -> 9600 bits/s 19200 -> 19200 bits/s 38400 -> 38400 bits/s 57600 -> 57600 bits/s 115200 -> 115200 bits/s 230400 -> 230400 bits/s 460800 -> 460800 bits/s 921600 -> 921600 bits/s	Default: 9600, 0, 0	<p>Set baud rate to be 115200, stop bit to be 2 bits, parity bit to be even parity.</p> <p>AT+UART=115200,1,2,r\n OK</p> <p>AT+UART?</p> <p>+UART:115200,1,2</p> <p>OK</p>
	AT+UART?	+UART=Param1, Param2, Param3 OK	1382400 -> 1382400 bits/s Param2: Stop bit: 0 -> 1 bit 1 -> 2 bits Param3: Parity bit: 0 -> None 1 -> Odd parity 2 -> Even parity		
Set/Inquire connection mode	AT+CMODE=Param1	OK	Param1: Connection mode: 0 -> Connect the module to the specified Bluetooth address. (Bluetooth address can be specified by the binding command) 1 -> Connect the module to any address (The specifying address has no effect for this mode.) 2 -> Slave-Loop	Default connection mode: 0	<p>Set the module device connection mode to Slave-Loop:</p> <p>AT+CMODE=2</p> <p>OK</p> <p>AT+CMODE?</p> <p>+CMODE:2</p> <p>OK</p>
	AT+ CMODE?	+CMODE:Param1 OK			
Set/Inquire – bind Bluetooth address	AT+BIND=Param1	OK	Param1: Bluetooth address: needed to be bind	<p>Bluetooth address will show as this way: NAP: UAP:LAP(Hexadecimal)</p> <p>This command is effective only when the module wants to connect to the specified Bluetooth address.</p> <p>Default: 00:00:00:00:00:00</p>	<p>The module is at connection mode which connects to specified Bluetooth address, and the specified address is 12:34:56:ab:cd:ef.</p> <p>AT+BIND=1234, 56, abcdef r\n OK</p> <p>AT+BIND? r\n</p> <p>+BIND:1234:56:abcdef</p> <p>OK</p>
	AT+ BIND?	+BIND:Param1 OK			

Software Design

1. TelosB Firmware
2. Shell BASH Boot and Dynamic Configuration
3. ADC
4. UART
5. Arduino Firmware
6. BlueCove Application
7. Spring Boot Application
8. MySQL
9. Web App UI
10. React
11. Next
12. Supporting Software
13. Web UI



TelosB Firmware



<https://github.com/tinyos/tinyos-main>

Powered By: Visual Paradigm Community Edition

Shell BASH Boot and Dynamic Configuration

```
#!/bin/bash
set +x

if [ "$#" -eq 0 ]; then
    echo "Usage: $0 <argument>"
    exit 1
fi

# Access the first argument using $1
arg1="$1"

source ./configMakefile.sh "$arg1"
echo "./configMakefile.sh arg=$arg1"
sleep 0.1

source ./configMainApp.sh "$arg1"
echo "./configMainApp.sh arg=$arg1"
sleep 0.1

source ./configMain.sh "$arg1"
echo "./configMain.sh arg=$arg1"
sleep 0.1

source ./configUARTdriver.sh "$arg1"
echo "./configUARTdriver.sh arg=$arg1"
sleep 0.1

source ./configWaterAnalysis.sh "$arg1"
echo "./configWaterAnalysis.sh arg=$arg1"
sleep 0.1

source ./configQualityParamC.sh "$arg1"
echo "./configQualityParamC.sh arg=$arg1"
sleep 0.1
```

```
root@instant-contiki: /opt/tinyos-main/apps/progetto-iot
File Edit View Search Terminal Help
user@instant-contiki:~$ sudo su
[sudo] password for user:
root@instant-contiki:/home/user# cd ../../
root@instant-contiki:# cd opt/tinyos-main/apps/progetto-iot
root@instant-contiki:/opt/tinyos-main/apps/progetto-iot# ls
analysis configMainApp.sh configQualityParamC.sh driver Makefile
boot.sh configMain.sh configUARTdriver.sh main
build configMakefile.sh configWaterAnalysis.sh MainApp.nc
root@instant-contiki:/opt/tinyos-main/apps/progetto-iot#
```



Shell BASH Boot and Dynamic Configuration

```
TelosBFirmware/configMainApp.sh

generate_components() {
    case $arg1 in
        "home")
            cat <<EOL >> "$MAINAPP_DIR"
            components TemperatureC;
            components new Msp430SparkfunTemperatureC() as TemperaturesensorDriver;
            components TDS;
            components new Msp430SparkfunTDSC() as TDSSensorDriver;
            components PHC;
            components new Msp430SparkfunPHC() as PHsensorDriver;
            EOL
                ;;
        "sea")
            cat <<EOL >> "$MAINAPP_DIR"
            components TemperatureC;
            components new Msp430SparkfunTemperatureC() as TemperaturesensorDriver;
            components TDS;
            components new Msp430SparkfunTDSC() as TDSSensorDriver;
            components PHC;
            components new Msp430SparkfunPHC() as PHsensorDriver;
            EOL
                ;;
        "pool")
            cat <<EOL >> "$MAINAPP_DIR"
            components TemperatureC;
            components new Msp430SparkfunTemperatureC() as TemperaturesensorDriver;
            components PHC;
            components new Msp430SparkfunPHC() as PHsensorDriver;
            EOL
                ;;
        *)
            echo "Unknown value in argument: $arg1"
            exit 1
            ;;
    esac
}

generate_components
```

```
TelosBFirmware/configQualityParamC.sh

generate_uses() {
    case $arg1 in
        "home")
            cat <<EOL >> "$QUALITYPARAMC_DIR"
            interface Temperature;
            interface TDS;
            interface PH;
            EOL
                ;;
        "sea")
            cat <<EOL >> "$QUALITYPARAMC_DIR"
            interface Temperature;
            interface TDS;
            interface PH;
            EOL
                ;;
        "pool")
            cat <<EOL >> "$QUALITYPARAMC_DIR"
            interface Temperature;
            interface PH;
            EOL
                ;;
        *)
            echo "Unknown value in argument: $arg1"
            exit 1
            ;;
    esac
}

generate_uses
```



TelosB Makefile

- Include MSP430 USART and printf library directories
- Define preprocessor macros (E.g: SAMPLING_TIME, NR_QUALITY_PARAMS, ...)
- Include project components directories (E.g: /main, /analysis/water, /driver, ...)

A screenshot of a terminal window titled "TelosBFirmware/Makefile". The window displays a portion of a Makefile with syntax highlighting for CFLAGS and include paths. The code includes definitions for preprocessor macros like DACK_DIM, DSOP, DEOP, and various sampling and quality parameters. It also lists include paths for project components like main, analysis, and water, along with specific parameter files for temperature, ADC, TDS, and PH. The file concludes with a standard include directive for makerules.

```
COMPONENT>MainApp

# Include directories for MSP430 USART and printf library
CFLAGS += -I$(TOSDIR)/chips/msp430/usart
CFLAGS += -I$(TOSDIR)/lib/printf

# Define preprocessor macros
CFLAGS += -DACK_DIM=2
CFLAGS += -DSOP=60
CFLAGS += -DEOP=62
CFLAGS += -DSOP_POS=0
CFLAGS += -DSAMPLING_TIME=1024
CFLAGS += -DUARTdata_DIM=9
CFLAGS += -DNR_QUALITY_PARAMS=3
CFLAGS += -DTEMPERATURE_POS=0
CFLAGS += -DTDS_POS=1
CFLAGS += -DPH_POS=2
CFLAGS += -DEOP_POS=7
# Include directories for project components
CFLAGS += -I./main
CFLAGS += -I./analysis
CFLAGS += -I./analysis/water
CFLAGS += -I./analysis/water/quality.param
CFLAGS += -I./analysis/water/quality.param/temperature
CFLAGS += -I./analysis/water/quality.param/temperatureadc
CFLAGS += -I./analysis/water/quality.param/tds
CFLAGS += -I./analysis/water/quality.param/tdsadc
CFLAGS += -I./analysis/water/quality.param/ph
CFLAGS += -I./analysis/water/quality.param/phadc
CFLAGS += -I./driver

include $(AKERULES)
```

<https://github.com/tinyos/tinyos-main>



TelosB MainApp

- /tinyos-main/tos/system/TimerMilliC.nc
- /tinyos-main/tos/system/LedsC.nc
- /tinyos-main/tos/lib/printf/PrintfC.nc
- /tinyos-main/tos/lib/printf/SerialStartC.nc
- /tinyos-main/tos/chips/msp430/usart/Msp430Uart0C.nc

```
components MainC;
components Main;
components new TimerMilliC() as Timer;
components LedsC;
components PrintfC, SerialStartC;
components UARTdriver;
components new Msp430Uart0C();
components WaterAnalysis;
components QualityParamC;
components TemperatureC;
components new Msp430SparkfunTemperatureC() as TemperaturesensorDriver;
components TDSC;
components new Msp430SparkfunTDSC() as TDSsensorDriver;
components PHC;
components new Msp430SparkfunPHC() as PHsensorDriver;
```

```
Main.Boot -> MainC;
Main.Leds -> LedsC;
Main.TimerSampling -> Timer;
Main.Driver -> UARTdriver;
Main.Analysis -> WaterAnalysis;
Main.TimerDriver -> Timer;
WaterAnalysis.Leds -> LedsC;
WaterAnalysis.QualityParam -> QualityParamC;
UARTdriver.Resource -> Msp430Uart0C;
UARTdriver.UartStream -> Msp430Uart0C;
UARTdriver.Msp430UartConfigure <- Msp430Uart0C;
UARTdriver.Leds -> LedsC;
QualityParamC.Temperature -> TemperatureC;
TemperatureC.Temperaturemeasure -> TemperaturesensorDriver.TemperaturereadInterface;
QualityParamC.TDS -> TDSC;
TDSC.TDMeasure -> TDSsensorDriver.TDSreadInterface;
QualityParamC.PH -> PHC;
PHC.PHmeasure -> PHsensorDriver.PHreadInterface;
```

<https://github.com/tinyos/tinyos-main>



TelosB Main

- Dynamic array: QualityParams
- Sampling every SAMPLING_TIME
- UART sending every 2*SAMPLING_TIME

```
● ● ● TelosBFirmware/main/Main.nc

/**
 * @var QualityParams
 * @desc Global array to store quality parameters.
 */
uint16_t QualityParams[NR_QUALITY_PARAMS] = {-1, -1, -1};

/**
 * @event TimerSampling.fired
 * @desc Event triggered when the sampling timer fires.
 */
event void TimerSampling.fired() {
    // Obtain quality parameters from the analysis interface.
    call Analysis.get(QualityParams);

    // Start the periodic timer for the subsequent data transmission.
    call TimerDriver.startPeriodic(2 * SAMPLING_TIME);
}

/**
 * @event TimerDriver.fired
 * @desc Event triggered when the data transmission timer fires.
 */
event void TimerDriver.fired() {
    // Send quality parameters through the Driver communication interface.
    call Driver.send(QualityParams);
}
```

ADC



```
● ● ● TelosBFirmware/analysis/water/quality.param/ph/Msp430SparkfunPHC.nc

/*
 * @file Msp430SparkfunPHC.nc
 * @brief Implementation of the generic configuration Msp430SparkfunPHC for pH sensor in
nesC on TinyOS.
 * @author [gubbriaco, fnicoletti, agrandinetti]
 */
generic configuration Msp430SparkfunPHC() {
    provides {
        interface Read<uint16_t> as PHreadInterface;
    }
}

implementation {
    components new AdcReadClientC() as ClientPH;

    PHreadInterface = ClientPH;

    components Msp430SparkfunPHP;

    ClientPH.AdcConfigure -> Msp430SparkfunPHP.Sensor;
}
```

```
● ● ● TelosBFirmware/analysis/water/quality.param/ph/Msp430SparkfunPHC.nc

/*
 * @file Msp430SparkfunPHP.nc
 * @brief Implementation of the Msp430SparkfunPHP module for pH sensor in nesC on TinyOS.
 * @author [gubbriaco, fnicoletti, agrandinetti]
 */

#include <Msp430Adc12.h>

module Msp430SparkfunPHP {
    provides {
        interface AdcConfigure<const msp430adc12_channel_config_t*> as Sensor;
    }
}

implementation {
    /**
     * @var configSens
     * @desc Declaration of configuration settings for the pH sensor.
     */
    const msp430adc12_channel_config_t configSens = {
        inch: INPUT_CHANNEL_A1, //0
        sref: REFERENCE_AVcc_AVss, //0
        ref_2_5v: REFVOLT_LEVEL_2_5, //0
        adc12ssel: SHT_SOURCE_ACLK, //1
        adc12div: SHT_CLOCK_DIV_1, //0
        sht: SAMPLE_HOLD_4_CYCLES, //0
        sampcon_ssel: SAMPCON_SOURCE_SMCLK, //2
        sampcon_id: SAMPCON_CLOCK_DIV_1 //0
    };

    /**
     * @async_command
     * @desc Asynchronous command to obtain the configuration settings for the pH sensor.
     * @return Configuration settings for the TDS sensor.
     */
    async command const msp430adc12_channel_config_t* Sensor.getConfiguration() {
        return &configSens;
    }
}
```



UART - TelosB

- Sent Data: QualityParameters[NR_QUALITY_PARAMS]
- Received Data: ArduinoData[ACK_DIM]

```
● ● ●                                     TelosBFirmware/driver/UARTdriver.nc

/**
 * @command Driver.send
 * @desc Sends the quality parameters through the UART driver interface.
 * @param QualityParameters Array containing quality parameters to be transmitted.
 */
command void Driver.send(uint16_t QualityParameters[NR_QUALITY_PARAMS]) {
    // Populate DataUart array with quality parameters
    DataUart[2] = QualityParameters[TEMPERATURE_POS] >> 8;
    DataUart[1] = QualityParameters[TEMPERATURE_POS] & 0xff;
    DataUart[4] = QualityParameters[TDS_POS] >> 8;
    DataUart[3] = QualityParameters[TDS_POS] & 0xff;
    DataUart[6] = QualityParameters[PH_POS] >> 8;
    DataUart[5] = QualityParameters[PH_POS] & 0xff;

    // Request the resource before sending data
    call Resource.request();

    // Receive acknowledgment from Arduino
    ArduinoData = receivedData[1] << 8 | receivedData[0];
}
```



UART - TelosB

- SOP: Start of Packet
- EOP: End of Packet

```
● ● ● TelosBFirmware/driver/UARTdriver.nc

/** @event Resource.granted
 * @desc Event triggered when the resource is granted.
 */
event void Resource.granted() {
    // Transmission: Send data via UartStream and toggle the LED state
    DataUart[SOP_POS] = SOP;
    DataUart[EOP_POS] = EOP;
    if(call UartStream.send(DataUart, UARTdata_DIM) == SUCCESS) {
        call Leds.led0Toggle();
    }

    // Reception: Receive data via UartStream and toggle the LED state
    if(call UartStream.receive(receivedData, ACK_DIM) == SUCCESS) {
        call Leds.led2Toggle();
    }
}
```



UART - Arduino

```
● ● ● ArduinoFirmware/ArduinoFirmware.ino#readFromUART

// Read data from UART communication
while (uartSerial.available() > 0) {
    char inChar = uartSerial.read();
    if (inChar == SOP) {
        // Start of Packet marker detected
        index = 0;
        inData[index] = '\0';
        started = true;
        ended = false;
    } else if (inChar == EOP) {
        // End of Packet marker detected
        ended = true;
        break;
    } else {
        // Store the incoming character in the buffer
        if (index < LENGTH) {
            inData[index] = inChar;
            index++;
            inData[index] = '\0';
        }
    }
}
```

```
● ● ● ArduinoFirmware/ArduinoFirmware.ino#readFromUART

// Process received data
if (started && ended) {
    // Parse and update data array
    for (int i = 0; i < NR_SENSORS; i++) {
        data[i] = word(byte(inData[2 * i + 1]), byte(inData[2 * i]));
    }
    // Acknowledge successful data reception
    ack = 1;
    successfullyUART();
} else {
    // Acknowledge failure in data reception
    ack = 0;
    failureUART();
}
```



Arduino Firmware – Setup

```
AltSoftSerial uartSerial;  
  
SoftwareSerial bluetoothSerial(BT_RX_PIN, BT_TX_PIN);  
  
OneWire oneWireWaterTemperature(ONE_WIRE_BUS_WATER_TEMPERATURE);  
DallasTemperature waterTemperatureSensor(&oneWireWaterTemperature);  
  
.....  
  
void setup() {  
    Serial.begin(BAUD_RATE);  
    while (!Serial) {  
    }  
  
    uartSerial.begin(BAUD_RATE);  
    bluetoothSerial.begin(BAUD_RATE);  
  
    waterTemperatureSensor.begin();  
  
    .....  
}
```



Arduino Firmware – Temperature Average

```
● ● ● ArduinoFirmware/ArduinoFirmware.ino#getTemperatureAverage

/**
 * Calculates and returns the average temperature based on the accumulated temperature
values
 * and the count of temperature measurements.
 *
 * @param temperature The sum of temperature values.
 * @param countMeasures The count of temperature measurements.
 * @return The average temperature.
 */
float getTemperatureAverage(float temperature, int countMeasures) {
    // Calculate average temperature, TDS, and pH
    float temperatureAverage = temperature / countMeasures;
    return temperatureAverage;
}
```



Arduino Firmware – TDS Compensation and Average

```
...
ArduinoFirmware#getTotalDissolvedMetalsAverage

/**
 * Calculates and returns the compensated average Total Dissolved Solids (TDS) based on the
 * accumulated TDS values, the count of TDS measurements, and the corresponding
 * temperature.
 *
 * @param tds The sum of TDS values.
 * @param countMeasures The count of TDS measurements.
 * @param temperature The corresponding temperature for TDS compensation.
 * @return The compensated average TDS.
 */
float getTotalDissolvedMetalsAverage(float tds, int countMeasures, float temperature) {
    // Convert TDS readings to compensated values based on temperature
    float tdsAverage = tds / countMeasures;
    float averageVoltage = (tdsAverage * V_REF) / RESOLUTION;
    float compensationCoefficient = 1.0 + 0.02 * (temperature - 25.0);
    float compensationVoltage = averageVoltage / compensationCoefficient;
    float tdscompensated =
        133.42 * compensationVoltage * compensationVoltage * compensationVoltage -
        255.86 * compensationVoltage * compensationVoltage +
        857.39 * compensationVoltage
    ) * 0.5;
    return tdscompensated;
}
```

- V_REF: reference voltage of the TelosB ADC (2.5V)
- RESOLUTION: TelosB ADC resolution (4096)



Arduino Firmware – pH Average

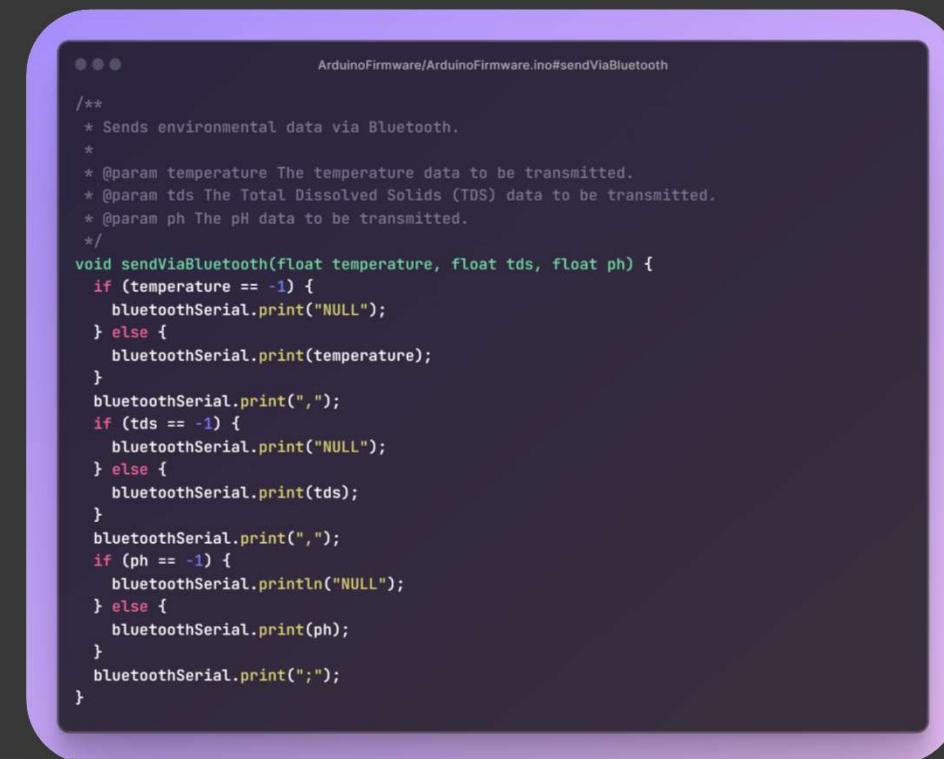
```
● ● ●
ArduinoFirmware/ArduinoFirmware.ino#getpHAverage

/**
 * Calculates and returns the average pH based on the accumulated pH values
 * and the count of pH measurements.
 *
 * @param ph The sum of pH values.
 * @param countMeasures The count of pH measurements.
 * @return The average pH.
 */
float getpHAverage(float ph, int countMeasures) {
    float phAverage = ph / countMeasures;
    return phAverage;
}
```



Arduino Firmware – Sending data to Base Station via Bluetooth

BluetoothDataPacket = temperature_value, tds_value, ph_value;



The screenshot shows the Arduino IDE interface with the following code:

```
ArduinoFirmware/ArduinoFirmware.ino#sendViaBluetooth

/*
 * Sends environmental data via Bluetooth.
 *
 * @param temperature The temperature data to be transmitted.
 * @param tds The Total Dissolved Solids (TDS) data to be transmitted.
 * @param ph The pH data to be transmitted.
 */
void sendViaBluetooth(float temperature, float tds, float ph) {
    if (temperature == -1) {
        bluetoothSerial.print("NULL");
    } else {
        bluetoothSerial.print(temperature);
    }
    bluetoothSerial.print(",");
    if (tds == -1) {
        bluetoothSerial.print("NULL");
    } else {
        bluetoothSerial.print(tds);
    }
    bluetoothSerial.print(",");
    if (ph == -1) {
        bluetoothSerial.println("NULL");
    } else {
        bluetoothSerial.print(ph);
    }
    bluetoothSerial.print(";");
}
```



BlueCove Application

```
FromBluetoothToServer/pom.xml

<dependencies>
    <!-- https://mvnrepository.com/artifact/io.ultreia/bluecove -->
    <dependency>
        <groupId>io.ultreia</groupId>
        <artifactId>bluecove</artifactId>
        <version>2.1.1</version>
    </dependency>
</dependencies>
```

```
FromBluetoothToServer/src/main/java/org.water.bluetooth/application/Main.java

public class Main {
    /**
     * The main method serves as the entry point for the Bluetooth server application.
     * It initializes and starts the Bluetooth server by obtaining an instance of the {@link
     * ServerApplication}.
     *
     * @param strings Command-line arguments (not used).
     */
    public static void main(String...strings) {
        // Create an instance of ServerApplication
        Server bs = ServerApplication.getInstance();
        // Start the Bluetooth server
        bs.start();
    }
}
```

<https://mvnrepository.com/artifact/io.ultreia/bluecove/2.1.1>



BlueCove Application - BluetoothServerExecutorApplication

```
● ● ●  FromBluetoothToServer/src/.../executor/BluetoothServerExecutorApplication.java#execute

@Override
public void execute() throws IOException {
    Logging.msg("Waiting for connections...");
    while (running) {
        // Create a new thread for each incoming connection
        Thread remoteConnection = new RemoteConnection(localDevice, notifier);
        remoteConnection.start();
    }
}
```

BlueCove Application – RemoteConnection

- Received message reading
- Received message checking
- JSON representation creation
- HTTP POST request

```
● ● ●  FromBluetoothToServer/src/.../executor/connection/RemoteConnection.java#run

InputStream inputStream = connection.openInputStream();
StringBuilder receivedData = new StringBuilder();
int character;

while ((character = inputStream.read()) != -1) {
    char receivedChar = (char) character;
    // The ; character at the end of the message is excluded.
    if (receivedChar != ';') {
        receivedData.append(receivedChar);
    }
    // If the current character is ;, then reading of the message is
    // finished, and the text can be displayed on the screen.
    String data;
    if (receivedChar == ';') {
        data = receivedData.toString().trim();
        // Split the received data into individual values
        String[] values = data.split(",");
        // Check if the array has at least three elements
        if (
            values.length >= 1 &&
            !values[0].equals("nan") &&
            !values[1].equals("nan") &&
            !values[2].equals("nan")
        ) {
            // Get current date and time
            LocalDateTime currentTime = LocalDateTime.now();
            DateTimeFormatter formatter = DateTimeFormatter.ofPattern(
                "yyyy-MM-dd HH:mm:ss"
            );
            String request= makeJsonString(values);
            HTTPPOST httppost= new HTTPPOST(request);
        } else {
            Logging.msg("Invalid data format: " + data);
        }
        // Reset buffer for the next message
        receivedData.setLength(0);
    }
}
```

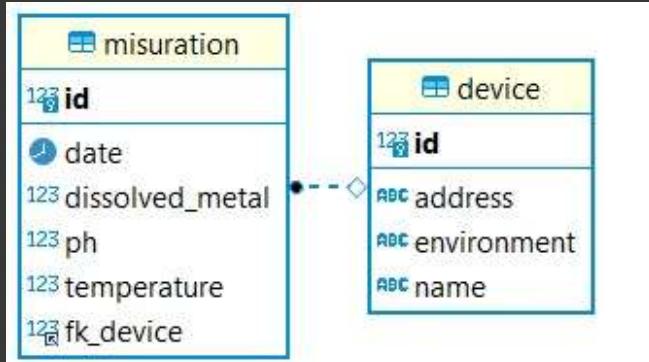


Spring Boot Application

- Java-based framework for building microservices and web apps
- Includes a built-in web server (such as Tomcat, Jetty, or Undertow) to simplify application deployment
- Integrated objectmapper which takes care of the serialization and deserialization of objects in JSON
- It provides a series of annotations with which it is possible to assign a role to the various classes
- There are several connection drivers called JDBC, compatible with most SQL or NOSQL databases
- It allows you to create and expose REST-type endpoints
- Provides all the tools to act as an ETL server

MySQL

123 id	ABC address	ABC environment	ABC name
1	98D331F6EE49	HOME	Rogliano
2	98D341F6E58C	POOL	Grimaldi
3	98D341F6DCF6	HOME	Settimo di Montalto



123 id	123 date	123 dissolved_metal	123 ph	123 temperature	123 fk_device
1	2024-01-10 11:10:22.795000	99,21	6,26	12,82	2
2	2024-01-10 12:18:38.966000	95,66	5,92	13,67	2
3	2024-01-10 12:57:31.756000	68,94	6,42	16,01	3
4	2024-01-10 13:01:25.873000	98,6	6,12	14,53	2
5	2024-01-10 13:35:22.608000	69,87	6,99	16,34	3
6	2024-01-10 13:39:21.839000	103,26	6,15	14,67	2
7	2024-01-10 14:13:14.612000	71,32	7,15	16,5	3
8	2024-01-10 14:17:18.188000	100,85	5,93	14,4	2
9	2024-01-10 14:51:06.633000	73,68	7,07	16,5	3
10	2024-01-10 14:55:14.668000	104,16	6,09	14,28	2
11	2024-01-10 15:28:58.648000	74,36	7,03	16,59	3
12	2024-01-10 15:33:10.864000	102,65	6,11	14,74	2
13	2024-01-10 16:06:50.670000	71	7,24	16,96	3
14	2024-01-10 16:11:07.328000	101,56	6	15,23	2
15	2024-01-10 16:44:42.660000	71,26	7,35	17	3
16	2024-01-10 16:49:03.639000	105,94	6,1	15,54	2
17	2024-01-10 17:22:34.683000	71,26	7,33	17	3
18	2024-01-10 17:26:55.950000	102,38	5,99	15,63	2
19	2024-01-10 18:00:26.693000	71,4	7,4	17	3
20	2024-01-10 18:04:52.282000	118,17	6,04	15,95	2
21	2024-01-10 18:27:41.056000	109,66	3,62	15,85	1
22	2024-01-10 18:38:19.696000	71,04	7,43	17	3
23	2024-01-10 18:42:48.628000	107,58	5,79	16,02	2

React

- React is an open-source JavaScript library, developed and maintained by Facebook
- It is based on a component programming model in which user interfaces are divided into self-contained, reusable components
- Introduces the concept of a "Virtual DOM"
- Introduces Hooks that allow you to use state and other React features within functional components
- TypeScript is an open-source programming language developed by Microsoft that extends JavaScript by adding optional static types, improving the maintainability and scalability of the code

```
src/frontend/app/root/team/page.tsx

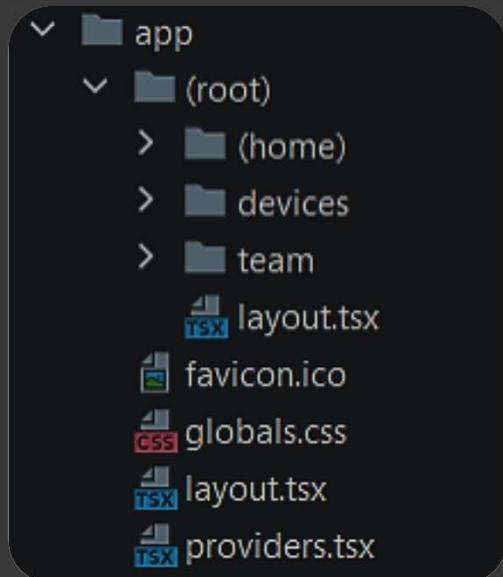
import { Avatar } from "@nextui-org/react";
import React from "react";
const teamData = [
  {
    name: "Amedeo Grandinetti",
    position: "CO-CTO",
    image: "./agra.png",
  },
  {
    name: "Giorgio Ubbriaco",
    position: "CO-CTO",
    image: "./gubb.png",
  },
  {
    name: "Francesco Nicoletti",
    position: "CO-CTO",
    image: "./fnic.png",
  },
];
const TeamPage = () => {
  return (
    <div className="grid grid-cols-3 gap-6 justify-between">
      {teamData.map((member, index) => (
        <div key={index} className="flex flex-col items-center">
          <Avatar src={member.image} className="h-56 w-56" />
          <div className="text-center mt-4">
            <h4 className="text-lg font-bold">{member.name}</h4>
            <p className="text-sm text-gray-600">{member.position}</p>
          </div>
        </div>
      )));
    );
}

export default TeamPage;
```



Next.js

Next.js is an open-source React framework that simplifies the development of performant and scalable web applications, providing features such as server-side rendering (SSR), static page generation, and a wide range of tools for state management and of navigation



```
src/frontend/components/header/Header.tsx#Header

<Navbar
  onMenuOpenChange={setIsMenuOpen}
  className="bg-primary text-white shadow-md"
>
  <NavbarContent>
    <NavbarMenuToggle
      aria-label={isMenuOpen ? "Close menu" : "Open menu"}
      className="sm:hidden"
    />
    <NavbarBrand>
      <code> WATER CONNECT</code>
    </NavbarBrand>
  </NavbarContent>

  <NavbarContent className="hidden sm:flex gap-6 justify-center">
    <NavbarItem isActive={pathname === "/" } className="text-lg">
      <Link href="/">Home</Link>
    </NavbarItem>
    <NavbarItem isActive={pathname === "/devices" } className="text-lg">
      <Link href="/devices" aria-current="page">
        Devices
      </Link>
    </NavbarItem>
    <NavbarItem isActive={pathname === "/team" } className="text-lg">
      <Link href="/team">Team</Link>
    </NavbarItem>
  </NavbarContent>
  <NavbarContent justify="end">
    <NavbarItem className="hidden lg:flex">
      <Link href="#">Login</Link>
    </NavbarItem>
    <NavbarItem>
      <Button as={Link} href="#" variant="bordered" className="text-white">
        Sign Up
      </Button>
    </NavbarItem>
  </NavbarContent>
  ....
</Navbar>
```



Supporting Software



HIBERNATE

*Maven*TM



PM2

node



Tailwind CSS



Server

```
ubuntu@vps-4bceaaca: $ sudo systemctl status api.service
● api.service - api
   Loaded: loaded (/etc/systemd/system/api.service; disabled; preset: enabled)
     Active: active (running) since Wed 2024-01-10 10:57:21 UTC; 13h ago
       Main PID: 31844 (java)
          Tasks: 32 (limit: 2247)
        Memory: 385.5M
         CPU: 2min 21.022s
        CGroup: /system.slice/api.service
                  └─31844 /bin/java -Xms512m -jar iotServer-0.0.1-SNAPSHOT.jar

Jan 11 00:33:24 vps-4bceaaca java[31844]: Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Jan 11 00:33:24 vps-4bceaaca java[31844]: Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Jan 11 00:33:32 vps-4bceaaca java[31844]: Hibernate: select d1_0.id,d1_0.address,d1_0.environment,d1_0.name from device d1_0
Jan 11 00:33:32 vps-4bceaaca java[31844]: Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Jan 11 00:33:32 vps-4bceaaca java[31844]: Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Jan 11 00:33:32 vps-4bceaaca java[31844]: Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Jan 11 00:33:39 vps-4bceaaca java[31844]: Hibernate: select d1_0.id,d1_0.address,d1_0.environment,d1_0.name from device d1_0
Jan 11 00:33:39 vps-4bceaaca java[31844]: Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Jan 11 00:33:39 vps-4bceaaca java[31844]: Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Jan 11 00:33:39 vps-4bceaaca java[31844]: Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
```



Linux system service

```
[Unit]
Description=api
After=syslog.target

[Service]
User=ubuntu
WorkingDirectory=/var/www/api
PIDFile=/run/waterConnect/api%i.pid
Restart=always
RestartSec=5
RuntimeDirectoryMode=755
Type=simple
#Process
ExecStart=/bin/java -Xms512m -jar iotServer-0.0.1-SNAPSHOT.jar

[Install]
WantedBy=multi-user.target
```



Maven

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.2.1</version>
    <relativePath/> 
  </parent>
  <groupId>it.progetto</groupId>
  <artifactId>iotServer</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>iotServer</name>
  <description>iotServer</description>
  <properties>
    <java.version>18</java.version>
    <maven.compiler.source>18</maven.compiler.source>
    <maven.compiler.target>18</maven.compiler.target>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jdbc</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
```



Hibernate

```
Hibernate: select d1_0.id,d1_0.address,d1_0.environment,d1_0.name from device d1_0
Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Hibernate: select d1_0.id,d1_0.address,d1_0.environment,d1_0.name from device d1_0
Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Hibernate: select d1_0.id,d1_0.address,d1_0.environment,d1_0.name from device d1_0
Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Hibernate: select d1_0.id,d1_0.address,d1_0.environment,d1_0.name from device d1_0
Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Hibernate: select d1_0.id,d1_0.address,d1_0.environment,d1_0.name from device d1_0
Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
Hibernate: select lm1_0.fk_device,lm1_0.id,lm1_0.date,lm1_0.dissolved_metal,lm1_0.ph,lm1_0.temperature from misuration lm1_0 where lm1_0.fk_device=?
```



PM2

```
ubuntu@vps-4bceaaca:/etc/systemd/system$ pm2 status frontend
```

id	name	namespace	version	mode	pid	uptime	�	status	cpu	mem	user	watching
0	frontend	default	N/A	fork	43860	4h	0	online	0%	66.9mb	ubuntu	disabled



Node.js

```
ubuntu@vps-4bceaaca:/etc/nginx/sites-enabled$ node
Welcome to Node.js v21.2.0.
Type ".help" for more information.
> -help
Uncaught ReferenceError: help is not defined
> .help
.break      Sometimes you get stuck, this gets you out
.clear      Alias for .break
.editor     Enter editor mode
.exit       Exit the REPL
.help       Print this help message
.load       Load JS from a file into the REPL session
.save       Save all evaluated commands in this REPL session to a file

Press Ctrl+C to abort current expression, Ctrl+D to exit the REPL
>
```



NPM

```
ubuntu@vps-4bceaaca:/etc/nginx/sites-enabled$ npm
npm <command>

Usage:
  npm install      install all the dependencies in your project
  npm install <foo> add the <foo> dependency to your project
  npm test         run this project's tests
  npm run <foo>    run the script named <foo>
  npm <command> -h  quick help on <command>
  npm -l           display usage info for all commands
  npm help <term>  search for help on <term>
  npm help npm     more involved overview

All commands:

  access, adduser, audit, bugs, cache, ci, completion,
  config, dedupe, deprecate, diff, dist-tag, docs, doctor,
  edit, exec, explain, explore, find-dupes, fund, get, help,
  hook, init, install, install-ci-test, install-test, link,
  ll, login, logout, ls, org, outdated, owner, pack, ping,
  pkg, prefix, profile, prune, publish, query, rebuild, repo,
  restart, root, run-script, search, set, shrinkwrap, star,
  stars, start, stop, team, test, token, uninstall, unpublish,
  unstar, update, version, view, whoami

Specify configs in the ini-formatted file:
  /home/ubuntu/.npmrc
or on the command line via: npm <command> --key=value

More configuration info: npm help config
Configuration fields: npm help 7 config

npm@9.2.0 /usr/share/nodejs/npm
```



Nginx

```
server {
    listen 80;
    listen [::]:80;

    index index.html index.htm;
    server_name waterconnect.it;

    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    client_max_body_size 12M;
}
```

Web App UI

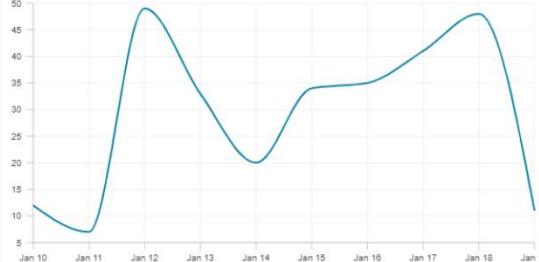
WATER CONNECT

Home Devices Team

Login Sign Up

Analysis of the device **Settimo di Montalto**

Settimo di Montalto



The graph displays two data series: Temperature (red line) and TDS (blue line). Both series show a sharp increase starting around Jan 11, peaking on Jan 12, and then fluctuating between Jan 13 and Jan 18 before ending on Jan 19.

Data	Temp.	TDS
Jan 10	16.01	68.94
Jan 11	16.34	69.87
Jan 12	16.5	71.32
Jan 13	16.5	73.68
Jan 14	16.59	74.36
Jan 15	16.96	71
Jan 16	17	71.26
Jan 17	17	71.26
Jan 18	17	71.4
Jan 19	17	71.04

Indirizzo: 98D341F6DCF6
Numero misurazioni: 16

Temp.	TDS	pH	Data	Ora
16.01	68.94	6.42	10/1/2024	13:57:31
16.34	69.87	6.99	10/1/2024	14:35:22
16.5	71.32	7.15	10/1/2024	15:13:14
16.5	73.68	7.07	10/1/2024	15:51:06
16.59	74.36	7.03	10/1/2024	16:28:58
16.96	71	7.24	10/1/2024	17:06:50
17	71.26	7.35	10/1/2024	17:44:42
17	71.26	7.33	10/1/2024	18:22:34
17	71.4	7.4	10/1/2024	19:00:26
17	71.04	7.43	10/1/2024	19:38:19



Web App UI

WATER CONNECT

Home Devices Team

Login Sign Up

Devices list +

Rogliano 1

Indirizzo: 98D331F6EE49
Numero misurazioni: 6

Grimaldi 2

Indirizzo: 98D341F6E58C
Numero misurazioni: 18

Settimo di Montalto 3

Indirizzo: 98D341F6DCF6
Numero misurazioni: 16

Progetto sviluppato da

WATER CONNECT

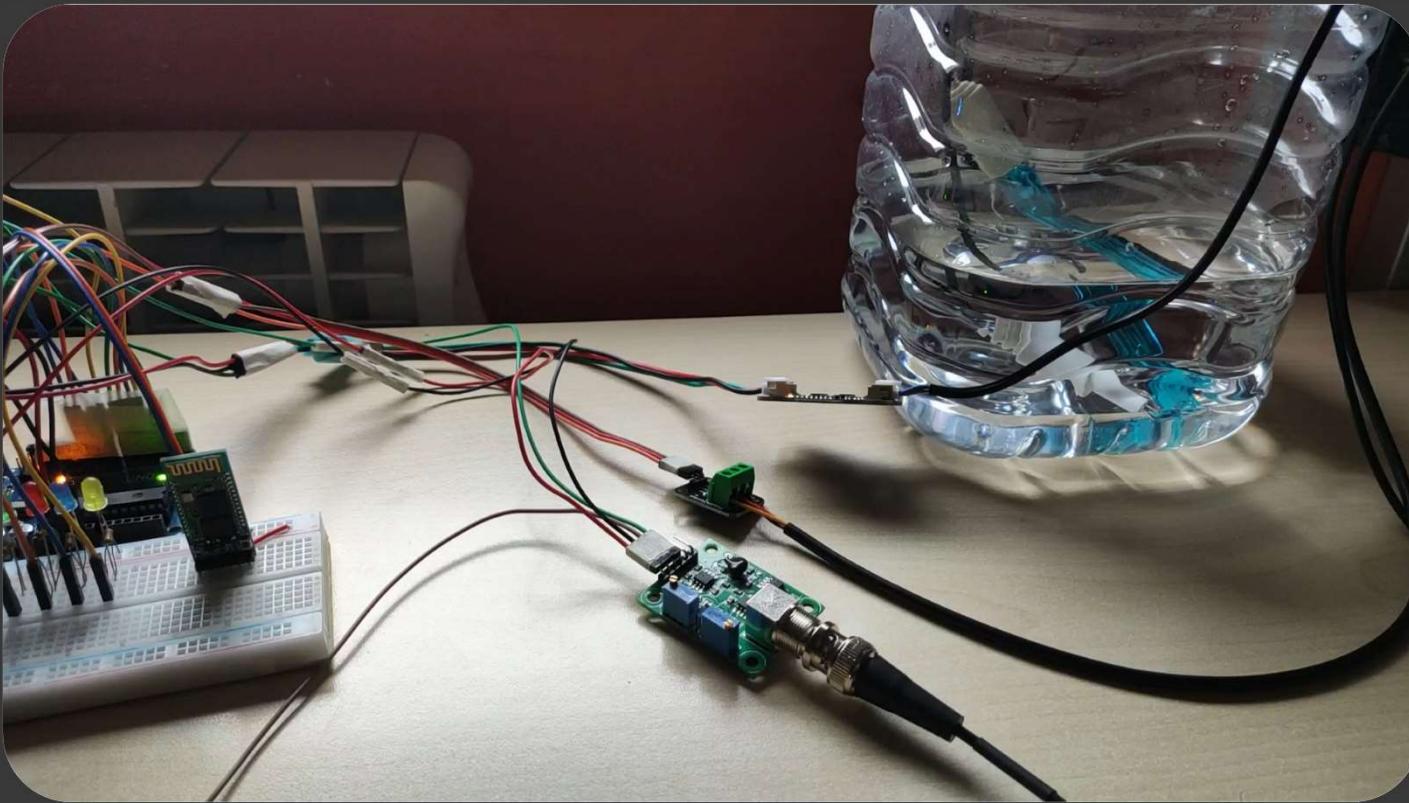
Home Devices Team

Login Sign Up

Grimaldi 2

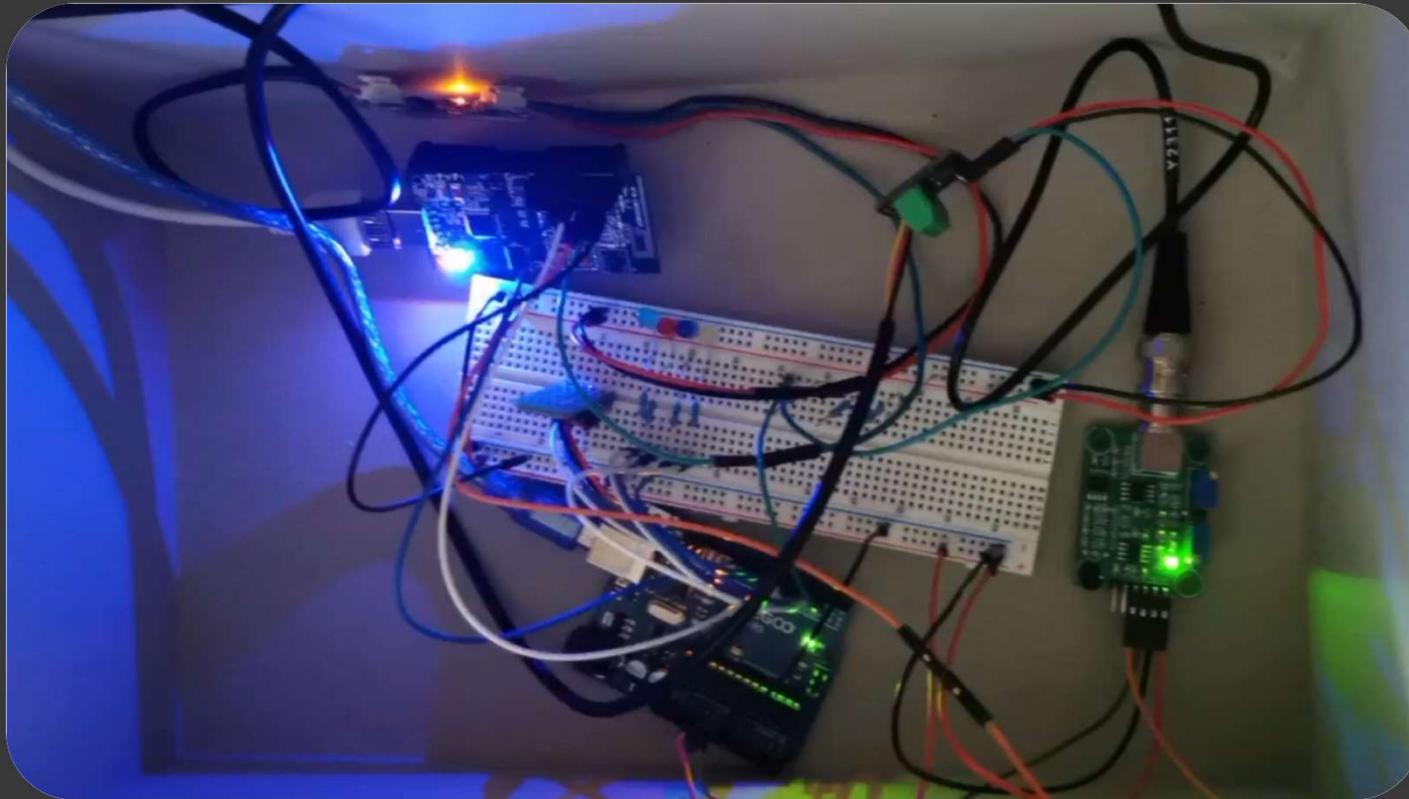
Temp.	TDS	pH	Data	Ora
12.82	99.21	6.26	10/1/2024	12:10:22
13.67	95.66	5.92	10/1/2024	13:18:38
14.53	98.6	6.12	10/1/2024	14:01:25
14.67	103.26	6.15	10/1/2024	14:39:21
14.4	100.85	5.93	10/1/2024	15:17:18
14.28	104.16	6.09	10/1/2024	15:55:14
14.74	102.65	6.11	10/1/2024	16:33:10
15.23	101.56	6	10/1/2024	17:11:07
15.54	105.94	6.1	10/1/2024	17:49:03
15.63	102.38	5.99	10/1/2024	18:26:55
15.95	118.17	6.04	10/1/2024	19:04:52
16.02	107.58	5.79	10/1/2024	19:42:48
15.9	104.88	5.75	10/1/2024	20:20:44
15.69	112	5.77	10/1/2024	20:58:41
15.45	117.95	6	10/1/2024	21:36:37
15.61	113.31	5.79	10/1/2024	22:14:33
15.97	117.06	6	10/1/2024	22:52:30
16.14	154.49	5.84	10/1/2024	23:30:26

Prototype – Settimo di Montalto Station



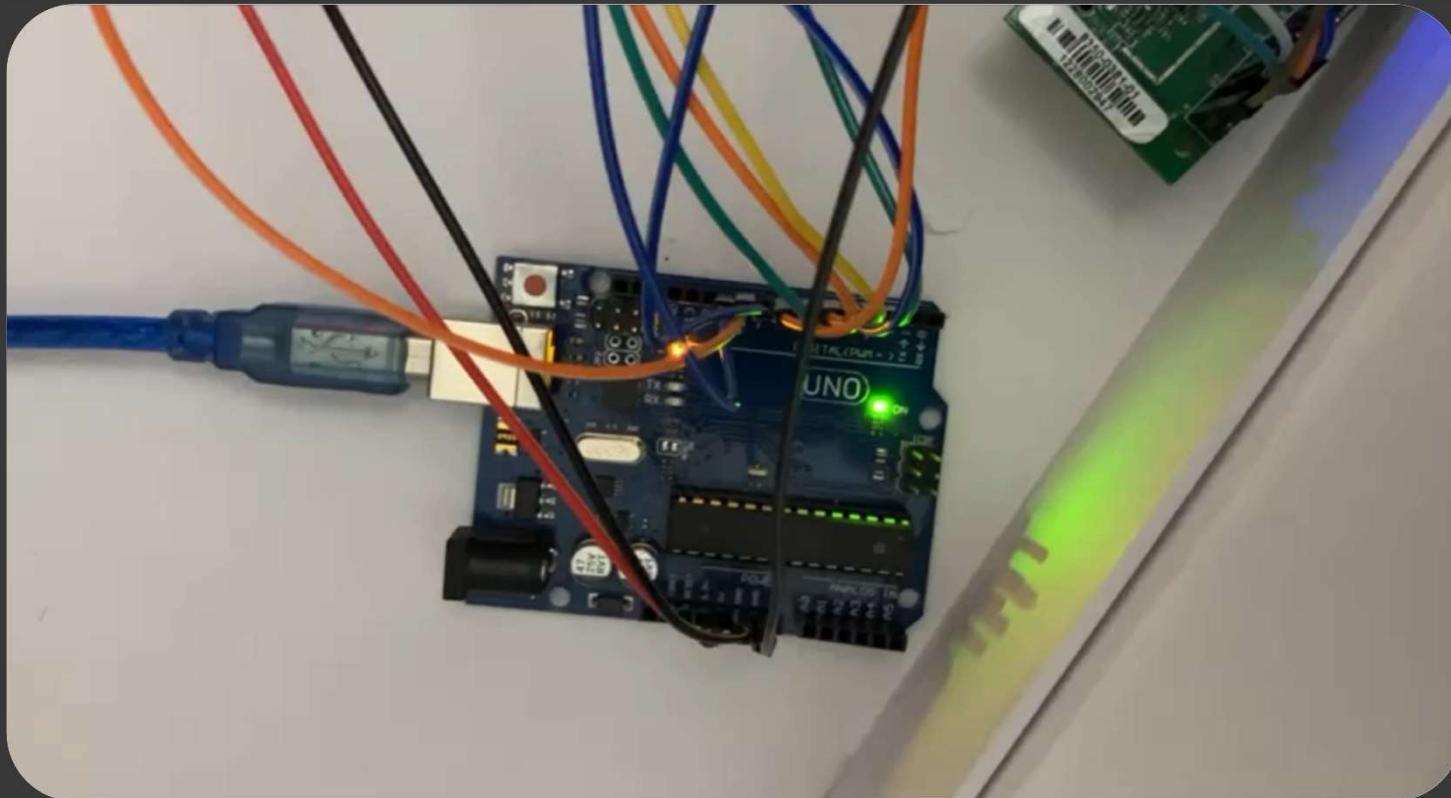


Prototype – Grimaldi Station

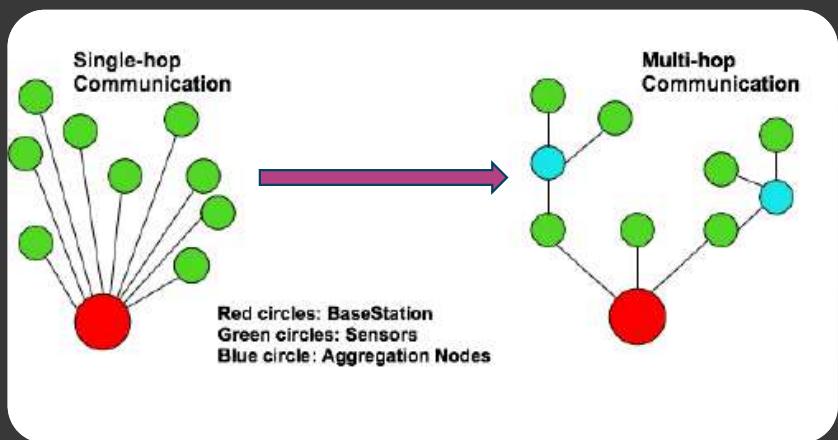




Prototype – Rogliano Station



Future Developments





Thanks!