SoRTES Lab - Session 2

Ashok Thangarajan, Stefanos Peros, Emekcan Aras

DistriNet

Recap(Session 1)-Arduino Bootloader

Serial

Arduino based board

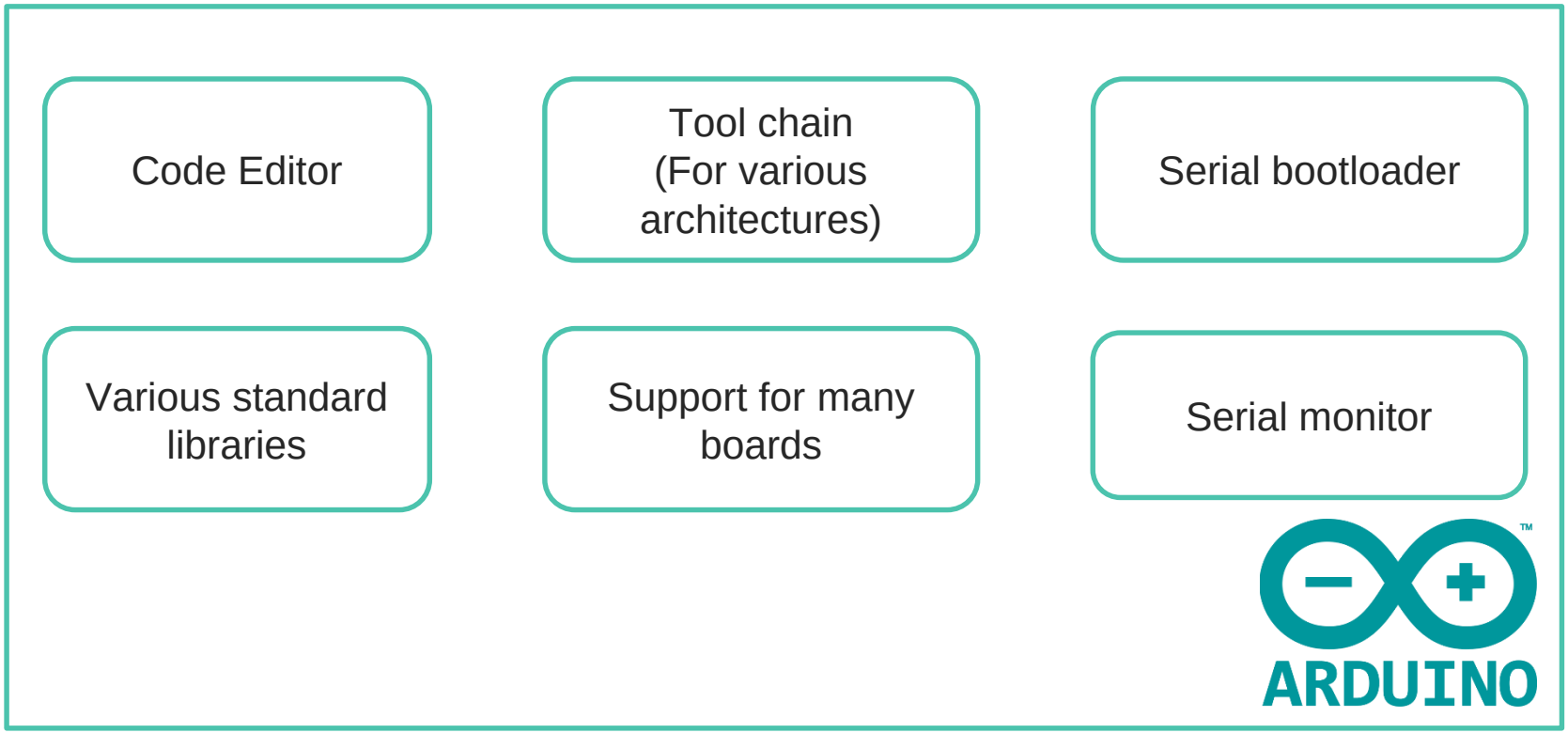
Flash memory

Serial bootloader

```
int main(void)
{
    init();

    initVariant();

#if defined(USBCON)
    USBDevice.attach();
#endif

    setup();

    for (;;) {
        loop();
        if (serialEventRun) serialEventRun();
    }
```


DistriNet

Recap(Session 1)-Arduino IDE



Recap(Session 1)Data Sheet

- Sometimes we have to read a Digital IO
  - How do we know if a pin is an IO pin or it is exposing an internal peripheral?
  - I2C timing characteristics?
  - How much current an IO pin can drive?

Recap(Session 1)-Sensors

- Sensing changes in physical elements – Sensors
- Sensors sense changes in environment
- Convert them to voltage levels
- Or covert them to processor's language
- Sensor output to microcontrollers:
  - I2C/SPI/UART
  - Analog output to processors ADC
  - Digital output via IO pins

Recap(Session 1)-Calibration

- What is calibration for sensors
- Slight offsets caused by:
  - Internal variations
  - Manufacturing variations
- How calibration is typically done

Recap(Session 1)-Flashing binary using avrdude – (1/2)

- Method to upload gateway binary to board
- Open blink example in Arduino from the following:
  - File->Examples->01.Basics->Blink
- Select File->Preferences, enable “Show verbose output during: Compilation and upload”
- Now click upload.
- Copy the upload command of avrdude

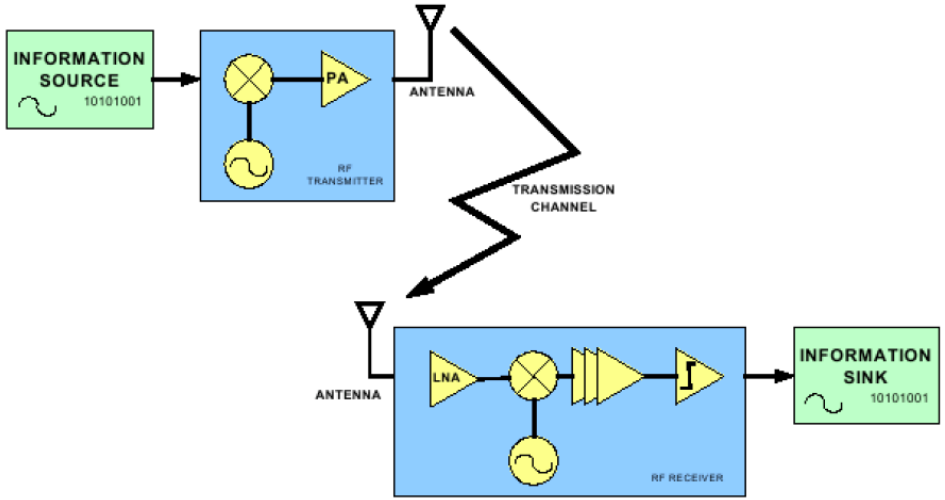
Recap(Session 1)-Flashing binary using avrdude – (2/2)

- Eg: C:\Users\XXUSER\Documents\ArduinoData\packages\arduino\tools\avrdude\6.3.0-arduino17/bin/avrdude -CC:\Users\XXUSER \Documents\ArduinoData\packages\arduino\tools\avrdude\6.3.0-arduino17/etc/avrdude.conf -v -patmega328p -carduino -PCOM38 -b57600 -D -Uflash:w:**C:\Users\XXUSER \AppData\Local\Temp\arduino_build_106667/Blink.ino.hex**:i
- Replace the line in bold with the path of the hex file.
- Open powershell in windows and run the command, it will upload the hex to the board

Session 2- Overview

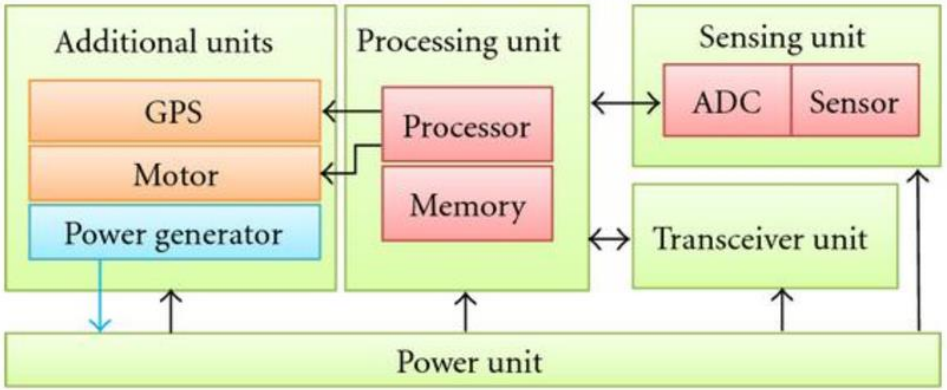
- Networked Embedded Devices in a Nutshell
- IoT Protocols
- LPWAN and LoRa
- Exercises

Network Embedded Devices in a Nutshell



- A transmitter and a receiver communicates through a wireless link.
- Embedded Device + Tramistter/receiver = Networked Embedded Device

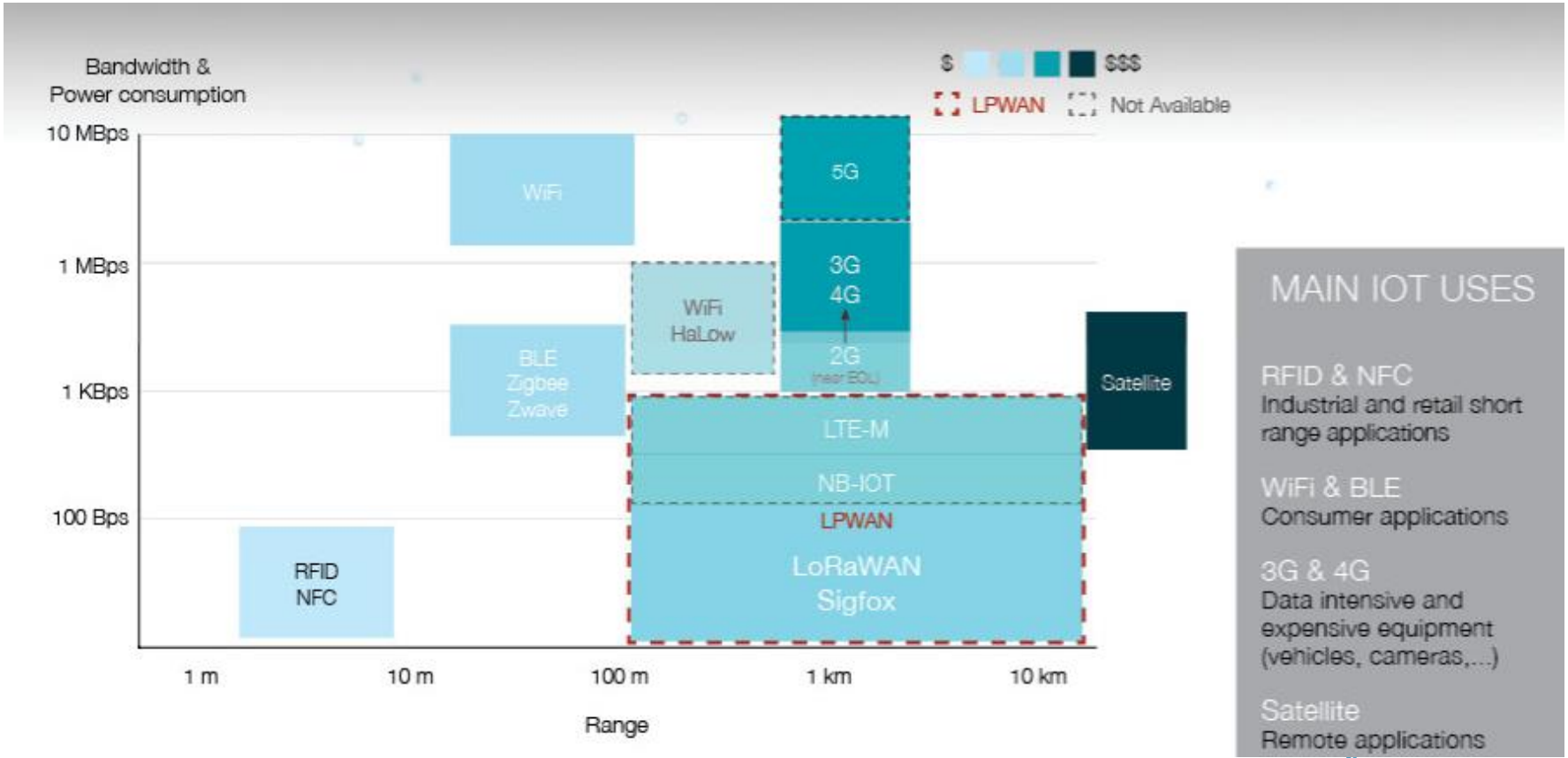
Network Embedded Devices in a Nutshell



GPS: global positioning system
ADC: analog to digital converter

- Different Hardware depending on the application
  - GPS
  - Sensors
  - Motor

Internet of Things and Embedded Devices



MAIN IOT USES

RFID & NFC
Industrial and retail short range applications

WiFi & BLE
Consumer applications

3G & 4G
Data intensive and expensive equipment (vehicles, cameras,...)

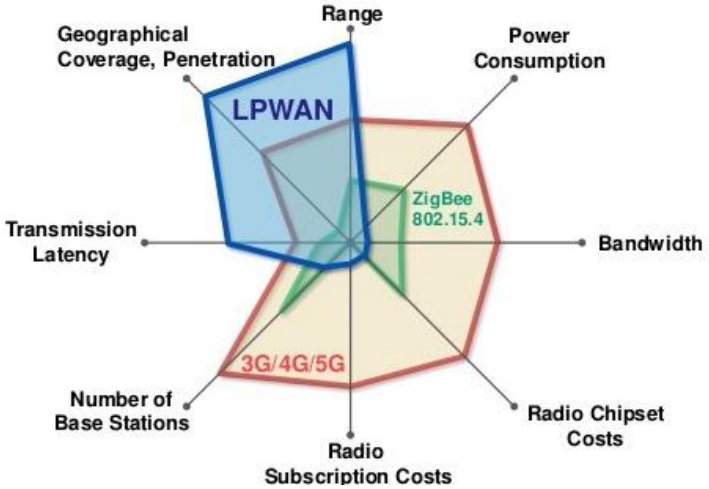
Satellite
Remote applications

DistriNet

What is LPWAN?

Low-Power Wide-Area Network (LPWAN) or Low-Power Network (LPN) is a type of wireless telecommunication network designed to allow long range communications at a low bit rate among things (connected objects), such as sensors operated on a battery

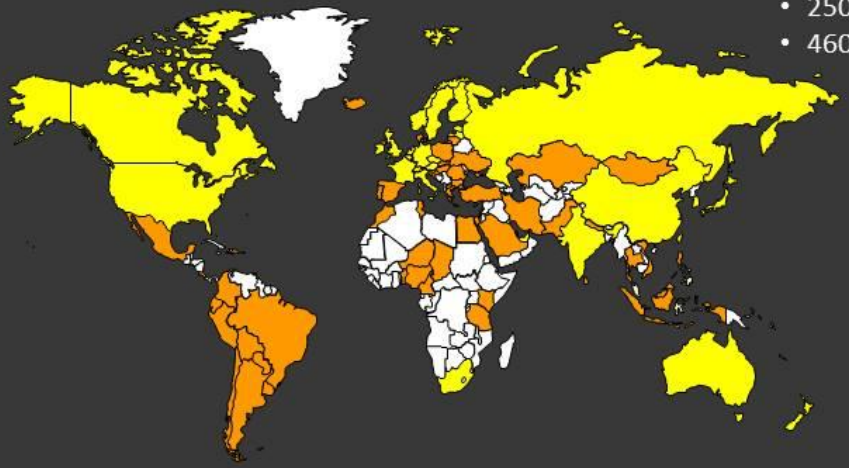
- LoRa
- SigFox
- NB-IOT
- NWave
- Weightless



DistriNet

Countries – LoRaWAN Networks

- 37 Publically Announced Operators
- 19 Alliance Member Operators
- 250+ on-going trials & city deployments
- 460+ members in the Alliance



Legend:

- Publicly Announced
- Other deployments

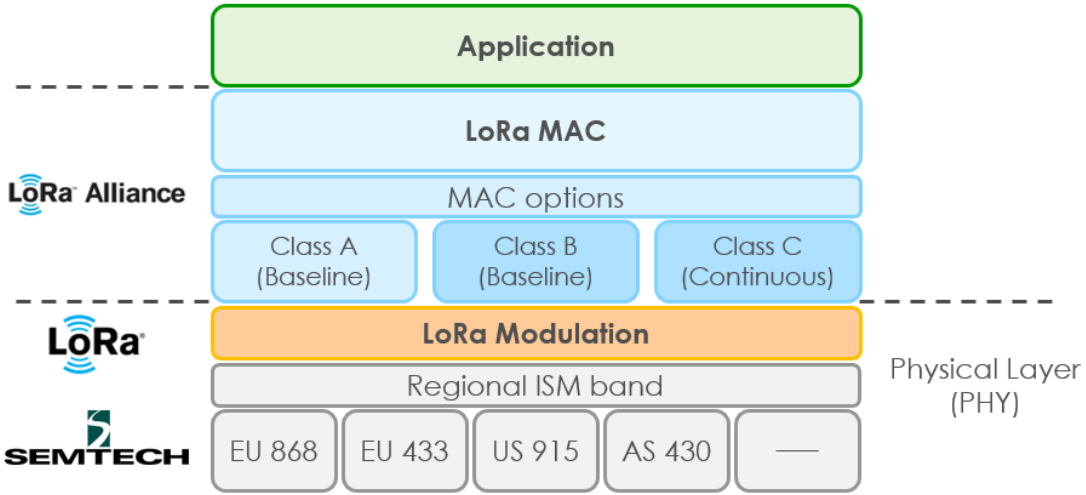
LoRa Alliance™

April 2017

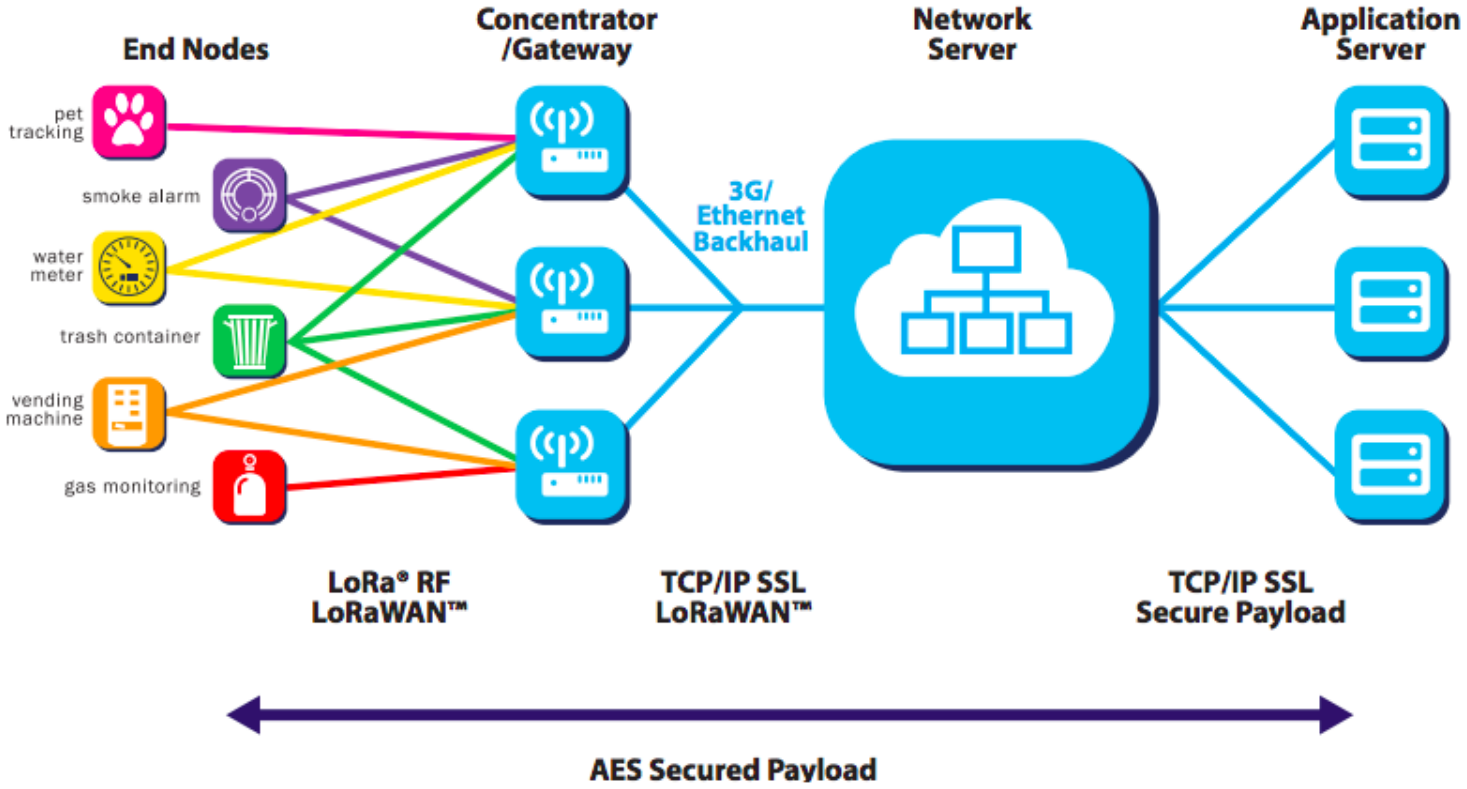
LoRa-Alliance.org

LoRa and LoRaWAN

- LoRa is the physical Layer
  - In EU, it operates 863-870 MHz
  - Multi km coverage
  - Low power consumptio
- LoRaWAN is the communication protocol

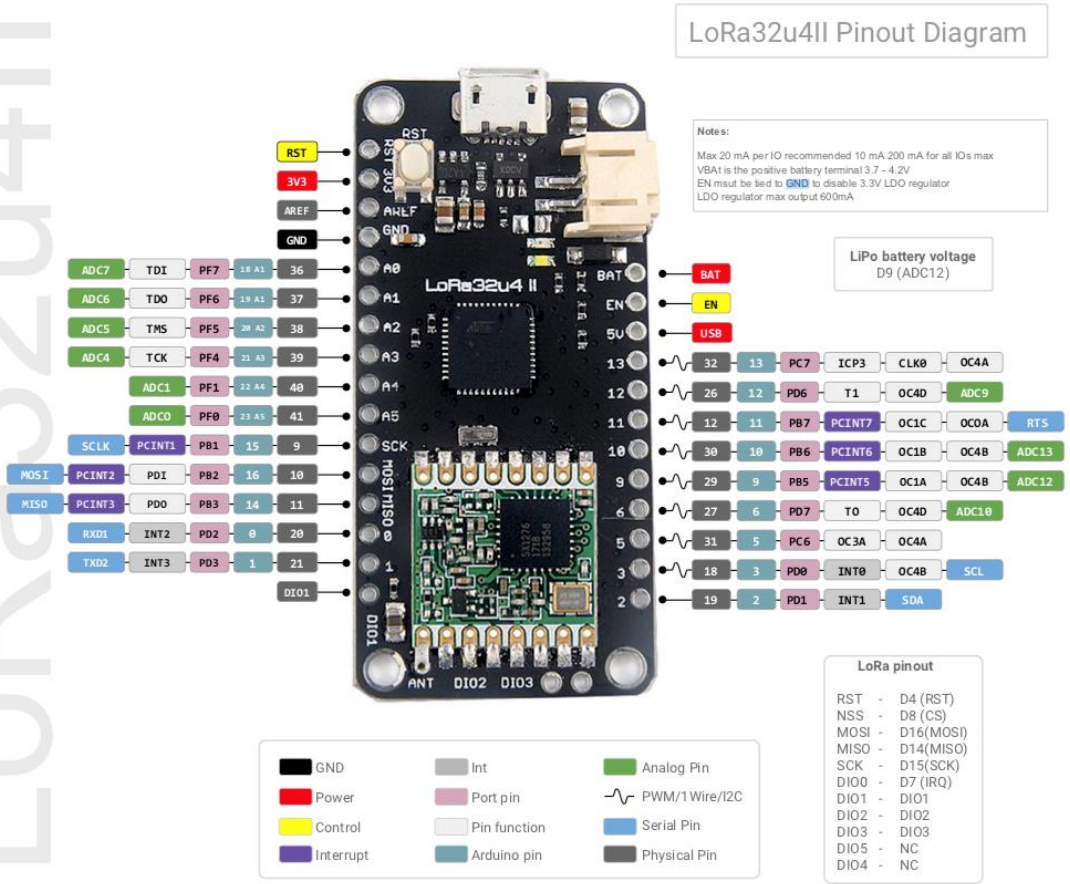


Typical LoRaWAN network



DistriNet

BsFrance Lora32u4 Board



LoRa32u4 Software

- Software Library inside BsFrance hardware directory
  - LoRa.cpp and LoRa.h
  - We will use basic LoRa functionality for the project

LoRa Basic - How to Setup the Radio module

- Base Frequency
  - frequency` - frequency in Hz (`433E6`, `866E6`, `915E6`)
- Spreading Factor (Sf)-(directly affects data-rate)
  - spreading factor, defaults to `7`
  - Supported values are between `6` and `12`.

LoRa Setup

- SPI communication
- LoRa frequency
- Serial module initialization

```
#define SCK     15
#define MISO    14
#define MOSI    16
#define SS      8
#define RST     4
#define DI0     7
#define BAND    868E6  // 915E6
#define PABOOST true

int counter = 0;

void setup() {
  Serial.begin(9600);
  while (!Serial);
  Serial.println("LoRa Sender");
  LoRa.setPins(SS,RST,DI0);
  if (!LoRa.begin(BAND,PABOOST)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
```

LoRa Send

- Consecutive Transmission
- Delay between next transmission

```
void loop() {
  Serial.print("Sending packet: ");
  Serial.println(counter);
  // send packet
  LoRa.beginPacket();
  LoRa.print("hello ");
  LoRa.print(counter);
  LoRa.endPacket();
  counter++;
  delay(5000);
}
```

LoRa Send Detail

› LoRa.write(byte), LoRa.print(byte);
  › Single byte
› LoRa.write(buffer, length);
  › Memcpy like function(for string write)
	To prepare the packet

› LoRa.endPacket() › Actually sends the packet	To send the packet

DistriNet

LoRa Receive

- Continuous Receive in a loop
- LoRa.parsePacket() checks the radio module for the new packet

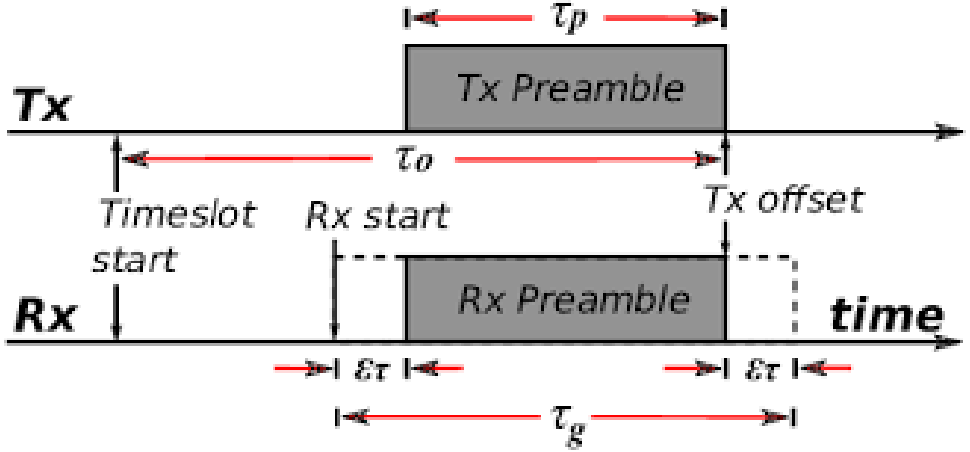
```
void loop() {
  // try to parse packet
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    // received a packet
    Serial.print("Received packet '");
    // read packet
    while (LoRa.available()) {
      Serial.print((char)LoRa.read());
    }
    // print RSSI of packet
    Serial.print("' with RSSI ");
    Serial.println(LoRa.packetRssi());
  }
}
```

LoRa Receive Detail

- LoRa.parsePacket();
  - Single packet receive mode.
  - LoRa.available()
    - To get the packets from radio module buffer(memory)
- LoRa.Receive();
  - Continuous receive mode. A callback needs to be initialized
    - LoRa.onReceive(callback);
      - Issue a callback(a function) to perform a procedure in case of a success message receive

Guard Time in Scheduled Transmission

- Don't have any time notion in LoRa
- Drifting on clock, turning on the transceiver and other operations causes a delay
- Guard-Time helps us to catch the scheduled transmission



Callback Example

- An example from the BsFrance Library
  - LoRaDuplexCallback

Code Examples

- BsFrance provides several examples
  - It’s under Arduino/Hardware/BsFrance/LoRa/examples
  - Check it README.md and API.md
  - Best way to understand how a module works is going through provided libraries and examples.

Exercises

Session 1 - Exercise 1

- Display the following in serial port (console), from your Arduino application
  - “Enter LED status (on/off):”
- Read the user input value from serial port into your application
- If value is “on”, display the following in serial port from your Arduino application
  - “Enter the blink rate (1-60 sec):”
- Print user provided values to serial port
  - “You have selected LED on/off. Blink rate is xx sec”
- After the user selected values are displayed in serial port, the blink pattern should be seen on the onboard LED
  - Info: Use a timer library to set the blink rate

Session 1 - Exercise 2

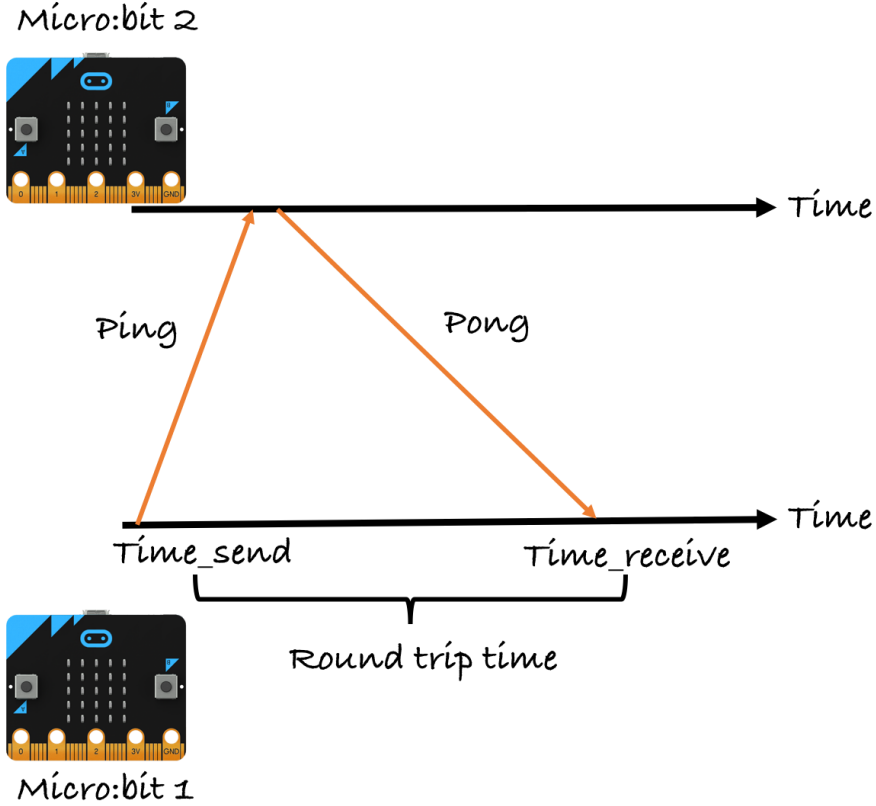
- Create a timestamp module (A simple counter should be enough)
- Create a small database (You may use Arduino library)
- Calibrate and read the temperature value from Arduino and store it into the database
- Power off and on the board, your database must be maintained without being deleted. (Look for Arduino EEPROM module)

Session2 - Exercise 1

- Run the Send-Receive example
  - Simple send and receive
    - One device sends the other receives
    - Change the message and send custom message with different intervals
    - Combine the temperature sensor exercise from previous week and send/receive temperature value from one device to another

Session2 - Exercise 2

- Ping Pong message passing
  - Write a software for both devices
    - A node start sending messages, the other one receives and sends the same message back to first node and first node receives the message and sends it back(. (Both device should send and receive messages consecutively)



Thank you!

https://distrinet.cs.kuleuven.be/