

# IT in AEC Industry 4.0

## IOT-NIMBUS

### n-Blocks & nBlocksStudio

### A Modular Low-Power Low-Code

### IoT

### Platform

Nikolaos Chalikias

Nimbus Center

MTU

[nikolaos.chalikias@mtu.ie](mailto:nikolaos.chalikias@mtu.ie)

# Contents

- IoT Hardware development in Nimbus
- n-Blocks Hardware Platform
  - Motivation
  - Concepts
  - Modularity
  - Some use cases
- nBlocksStudio Firmware Development Platform
  - Concepts
  - Workflow
  - Examples
- Summary

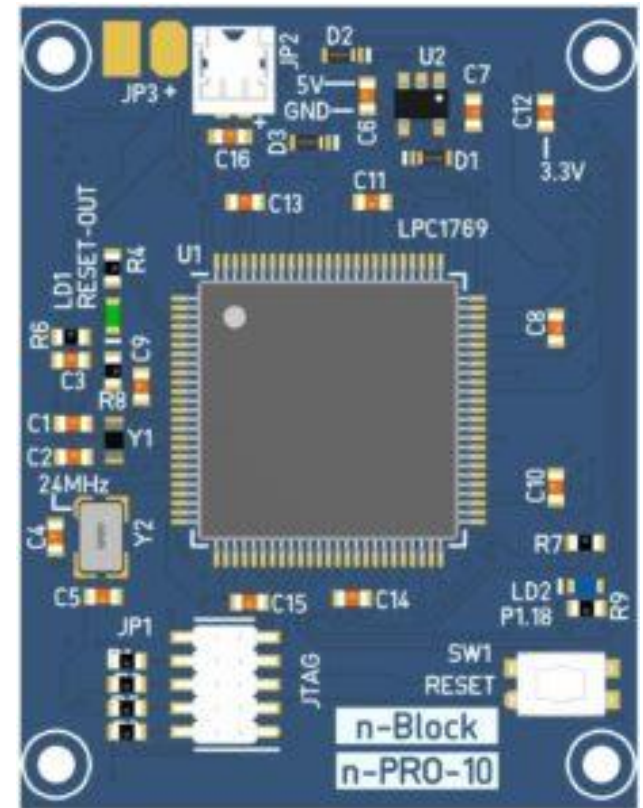
# IoT hardware development in Nimbus

## Embedded Hardware

PCB schematic and layout for rapid prototyping, proof of concept, pre-production prototypes

Simulation

Assembly, testing and verification.



# IoT hardware development in Nimbus

## Firmware

Communication protocols

Interface modules (SPI, UART, I2C, Smart GPIO...)

RF, Sensors, Remote Monitoring and Telemetry

Proximity, based on GPS, BLE and Wi-Fi.

Non-Blocking and Real time

```
#include "nbed.h"

PwmOut faclick(p21); // for RESERVE pin21 as PWM[4]
DigitalIn clikin(p30); // for RESERVE pin30 as CAP2[0]

// Reset Counter and Count Start
void P30_RESET_CTR(void)
{
    LPC_TIM2->TCR = 2; // Reset the counter (bits)
    LPC_TIM2->TCR = 1; // Unreset counter (bit1)
}

// Get Counter Value
int P30_GET_CTR(void)
{
    return LPC_TIM2->TC; // Read the counter value
}

// Setting p30 to Cap1.0
void P30_INIT_CTR(void)
{
    LPC_SC->PCONP |= 1 << 22; // 1)Power up TIM
    LPC_PINCON->PINSEL0 |= 3 << 8; // 2)Set PW[4] to
    LPC_TIM2->PCR = 2; // 3)Counter
    LPC_TIM2->CTCR = 1; // 4)count on rising edge
    LPC_TIM2->CCR = 0; // 4)count on rising edge
    LPC_TIM2->TCR = 1; // 5)Counter
}

// Clock Output from pin21(PWM)
// Set Clock Freq with div.
// If nbed is running at 96MHz, div is set 36 to get 1MHz.
void PWM_SETCLK(int div)
{
    LPC_PWM1->TCR = (1 << 1); // 1)Reset counter
    LPC_SC->PCLKSEL0 &= ~(0x3 << 12); // 2)Set peripheral
    LPC_SC->PCLKSEL0 |= (1 << 12); // 2)Set peripheral
```

# IoT hardware development in Nimbus

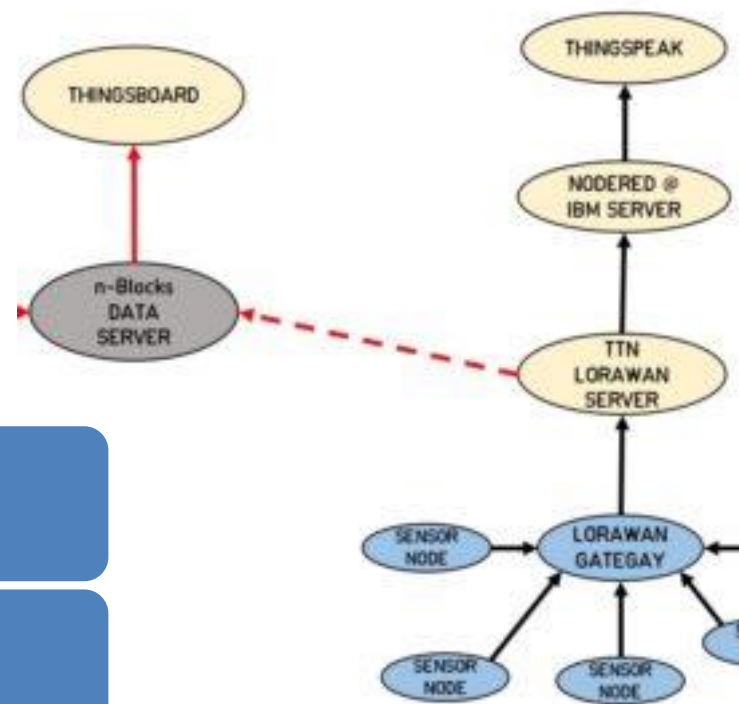
## System Integration

Interfacing the hardware and software components

System testing strategy

Validation of system-level functions throughout the integration process

Collaboration with Nimbus/CIT Software and Mechanical-design groups



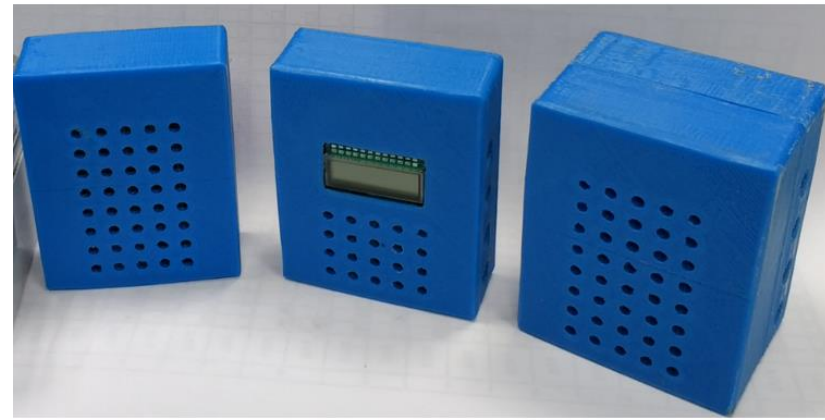
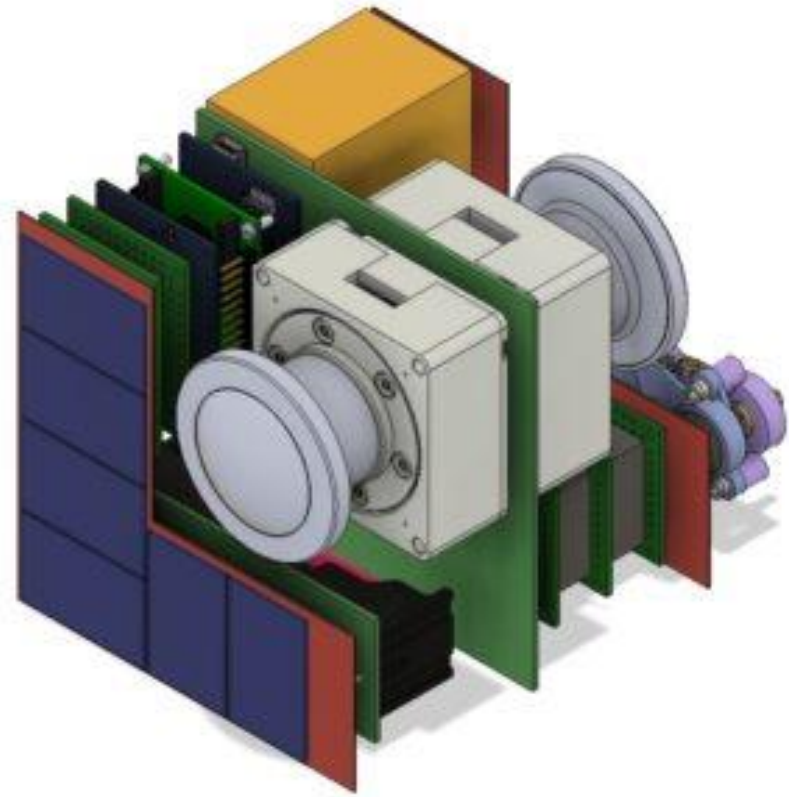
# IoT hardware development in Nimbus

## Product Design

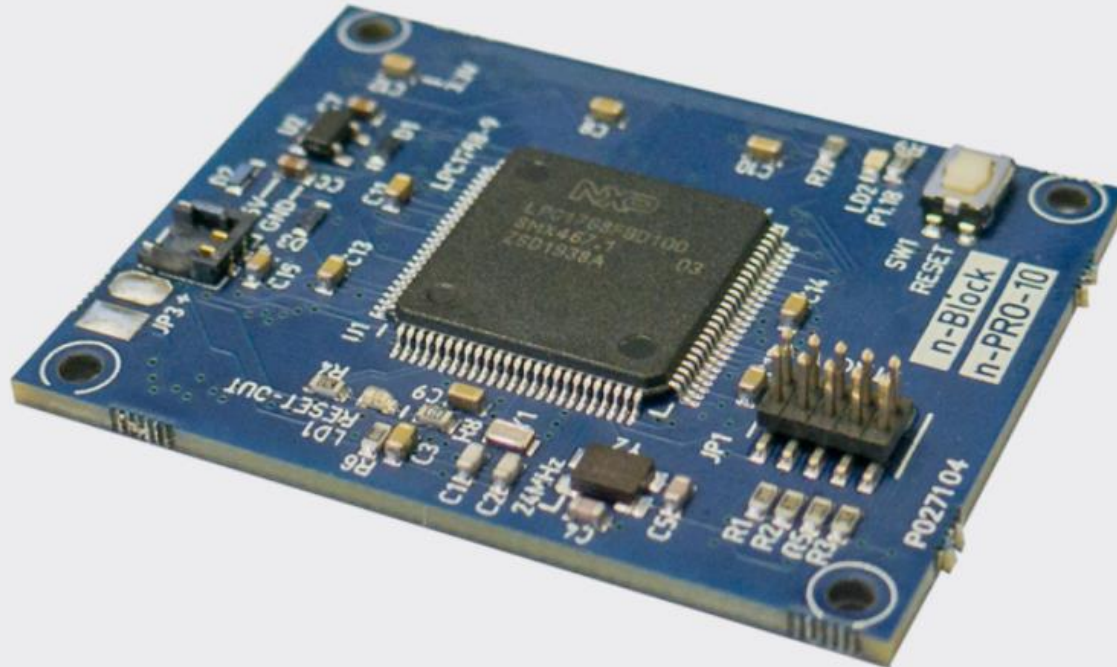
Overall system Architecture

Mechanical enclosure designs

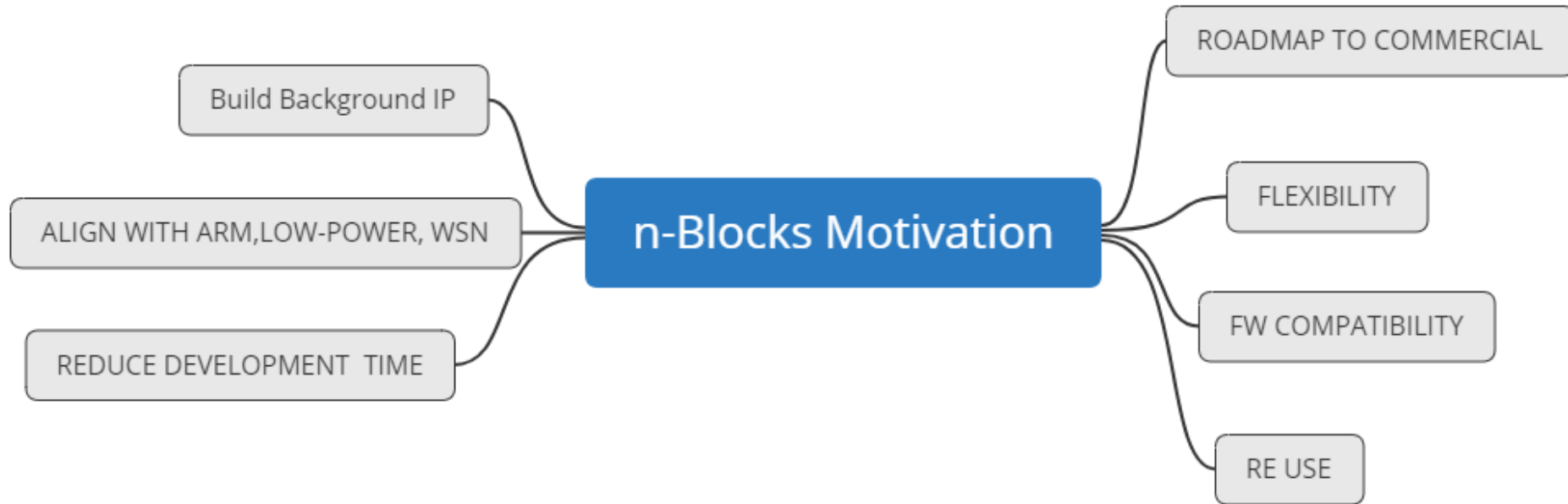
Small form factor designs



## n-Blocks Modular Platform for IoT Devices



# n-Blocks: Motivation



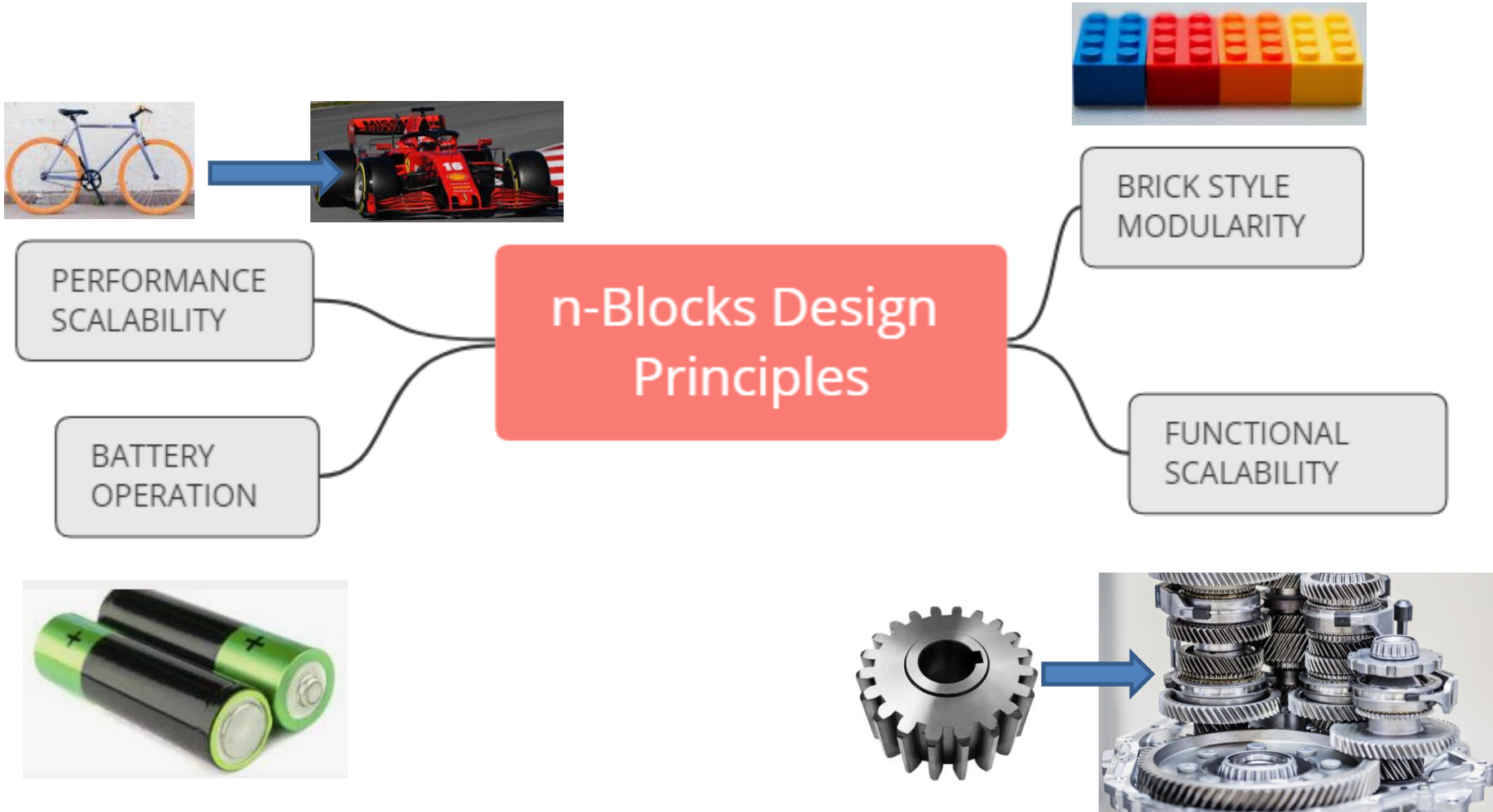
1. Provide an IoT proof of concept platform that offers an easy roadmap towards
  1. Commercial products
  2. Nimbus client's projects
2. Provide more flexibility than off-the-shelf technology
3. Compatibility with mainstream firmware development platforms
4. Re-use hardware/firmware
5. Reduce development time
6. Align development with ARM Cortex, Low Power & WSN
7. Build Background IP on hardware and firmware via standardization....OR.....



**HOW TO AVOID THE FUTURE DEVELOPERS TO TREAT OUR CURRENT WORK LIKE THIS  
HOW TO TRANSFORM OUR DAILY DEVELOPMENT TO BACKGROUND-IP**



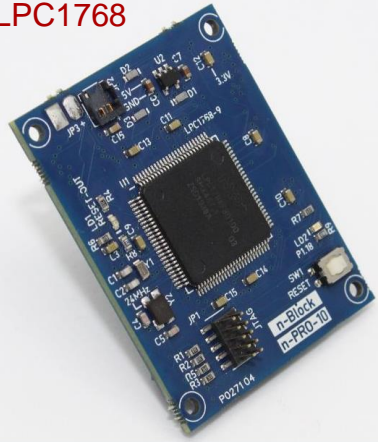
# n-Blocks: Modular IoT Design Principles





# n-Blocks: HW Modularity

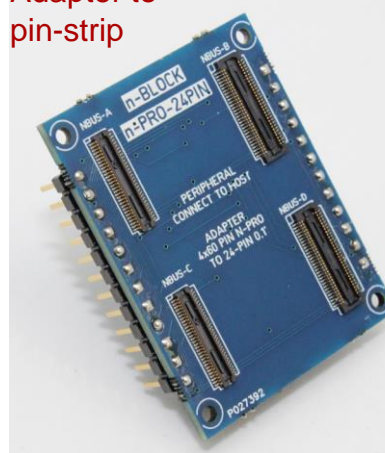
NXP Cortex M3  
LPC1768



TI Cortex A8  
OSD3358  
Linux



Adapter to  
pin-strip



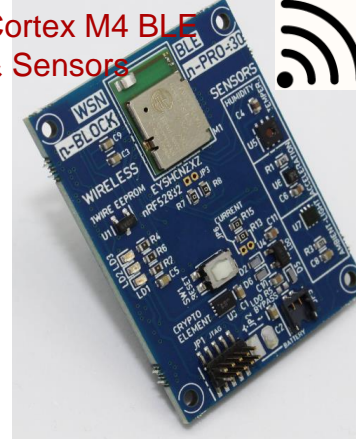
LoRa STM32L072  
SX1276 &  
Sensors



WIFI/BLT Espressif  
WROOM32  
& Sensors



BLE Nordic ARM  
Cortex M4 BLE  
& Sensors



Cortex M0, M3, M4  
STM32 64 pins

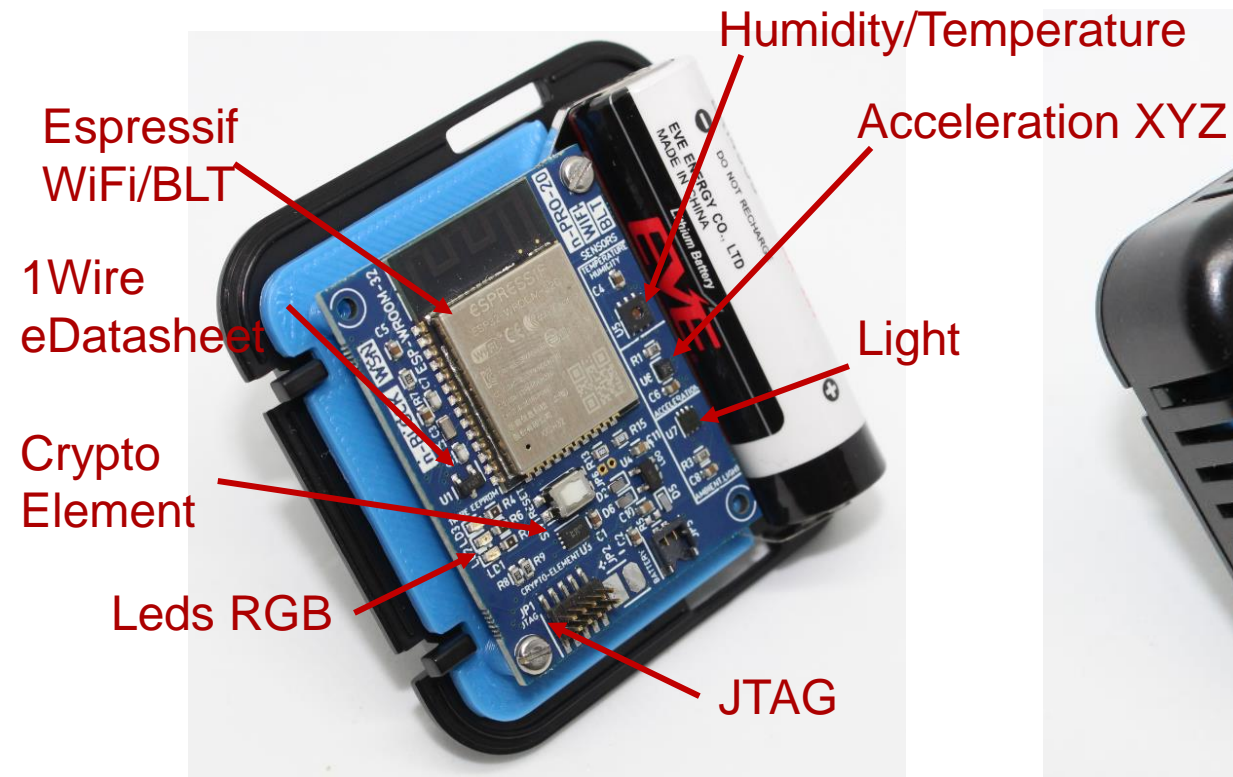


Wide  
MCU range

# Standalone Host Board : Wireless with sensors

## Example: n-PRO-20 WiFi / Host Sensor board

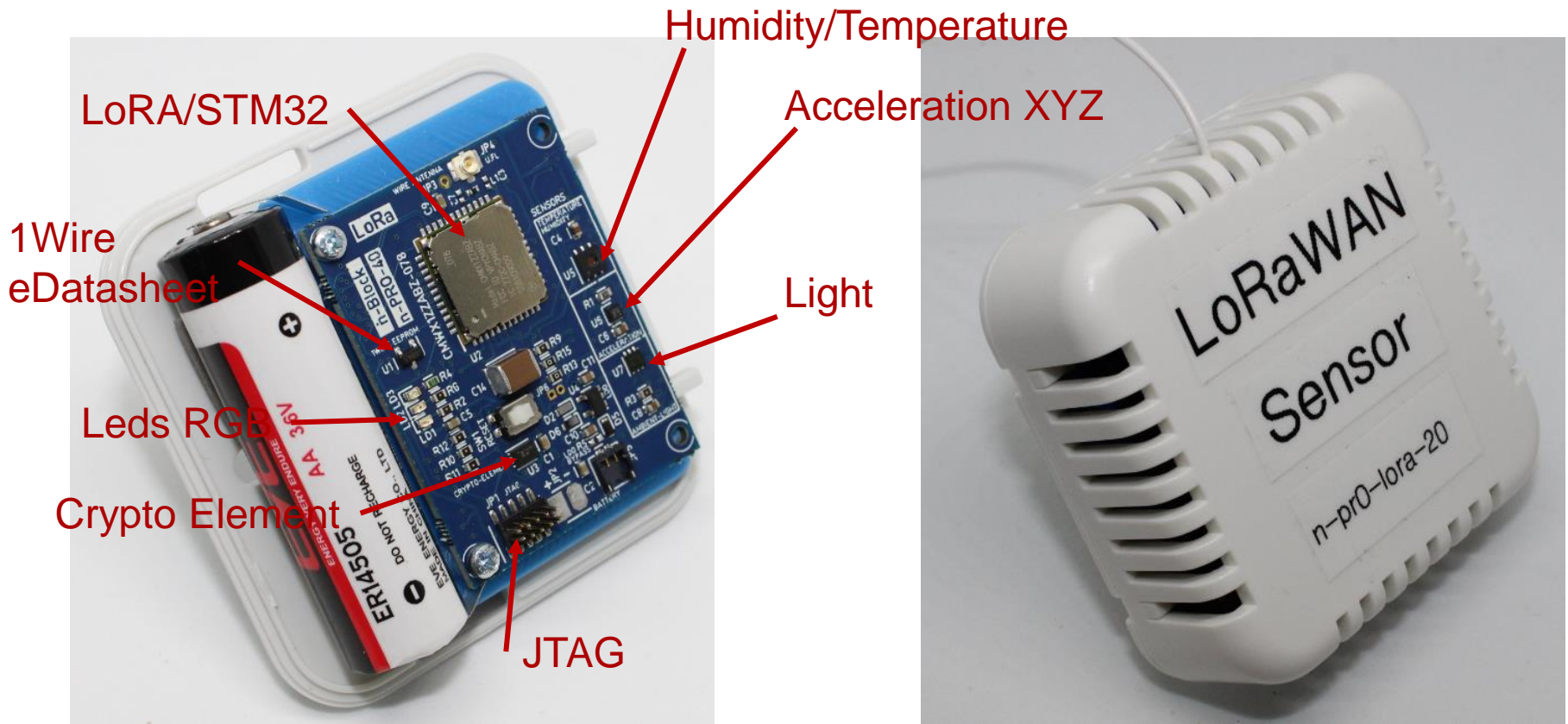
- Running on a 3 year battery life setup, on a low cost off-the-shelf indoor sensor enclosure



# Standalone Host Board: Wireless with Sensors

## Example: LoRa Sensor

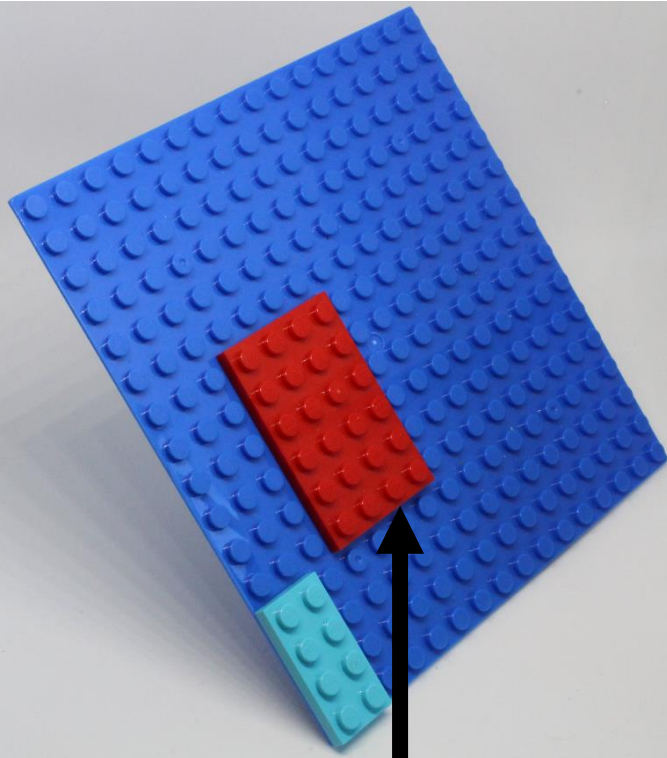
- Running LoRaWAN on a 10 year battery life setup, on a low cost off-the-shelf indoor sensor enclosure



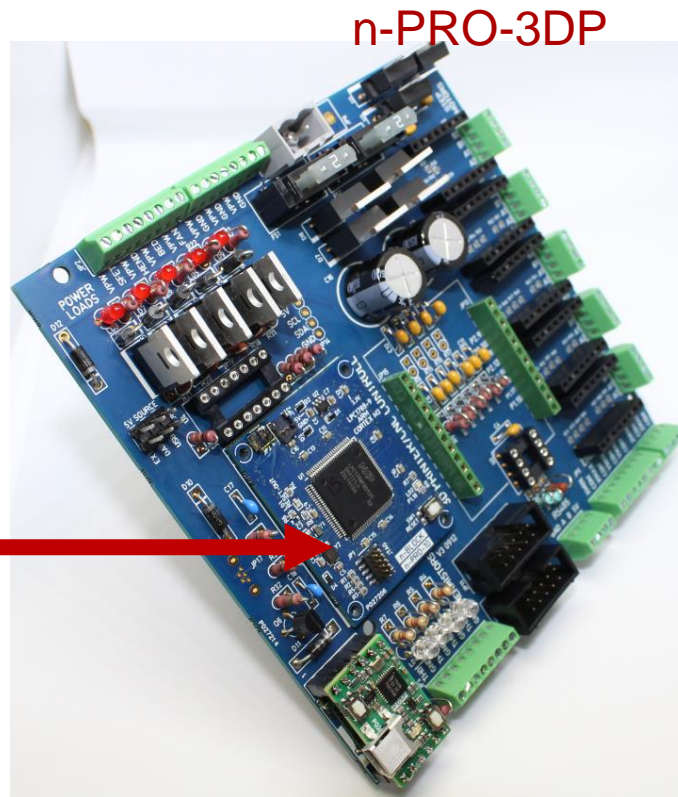


# Application Board + Host Board

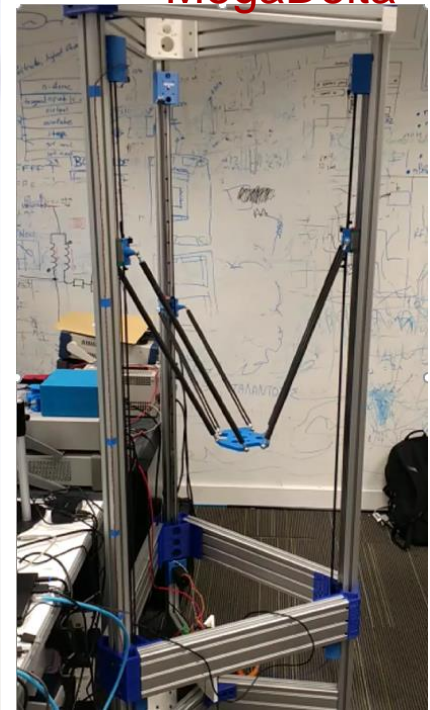
- Example: 3D printer / CNC / Robot controller
- Application board, with Host MCU board, tested on a MegaDelta Robot



n-PRO-10



n-PRO-3DP



MegaDelta

# n-Blocks: Wireless Sensor modularity

- Example: Feature-rich LoRa sensor
  - n-PRO-MBSEN Peripheral board VOC sensing, Magnetic contact, NFC, PIR motion, Sounder, expansion socket (120 pin) and AAA battery holder with low cost wall-mount indoors-sensor enclosure
  - n-PRO-40 LoRaWAN/sensors board
  - n-PRO-SEN1 peripheral board with ultra-low-power-I2C environmental sensors to expand sensing capabilities of a host node

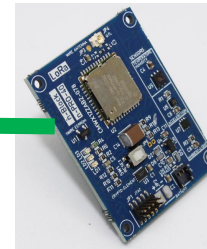
## n-PRO-SEN1 Ultra Low Power Sensors

- LIGHT
- PRESSURE
- VOC
- HUMIDITY
- ACCELERATION 3D



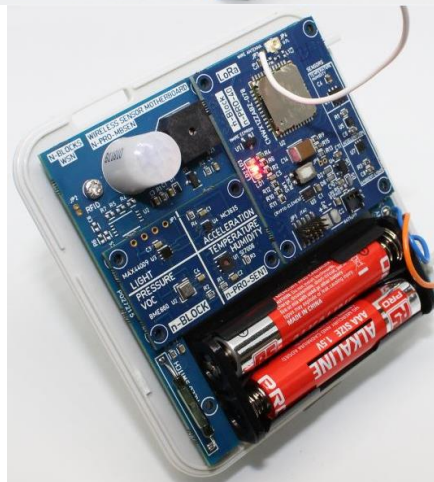
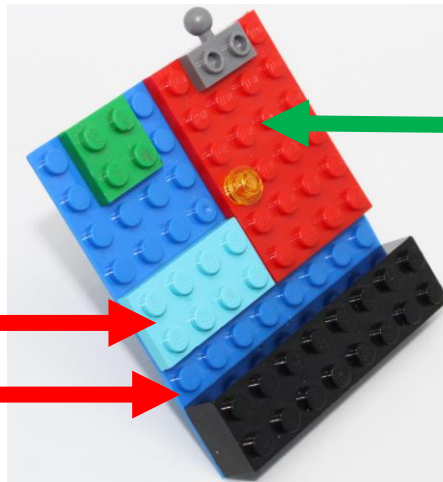
## n-PRO-40 LoRa n-Block

- LoRaWAN
- Crypto element
- 1 wire eeprom
- RGB leds
- JTAG
- On Board sensors



## n-PRO-MBSEN Wireless Sensor Motherboard

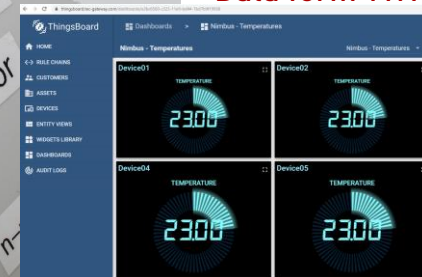
- PIR
- SOUNDER
- MAGNETIC CONTACT
- VOC
- HUMIDITY
- NFC
- SOUNDER



## ENCLOSURE COTS BMS Wall mount Sensor

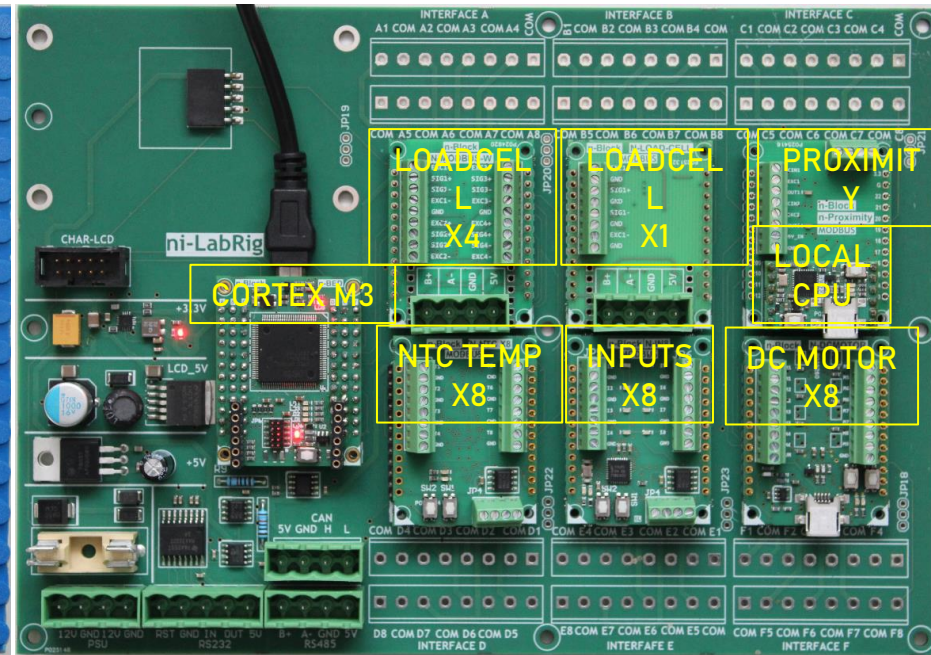
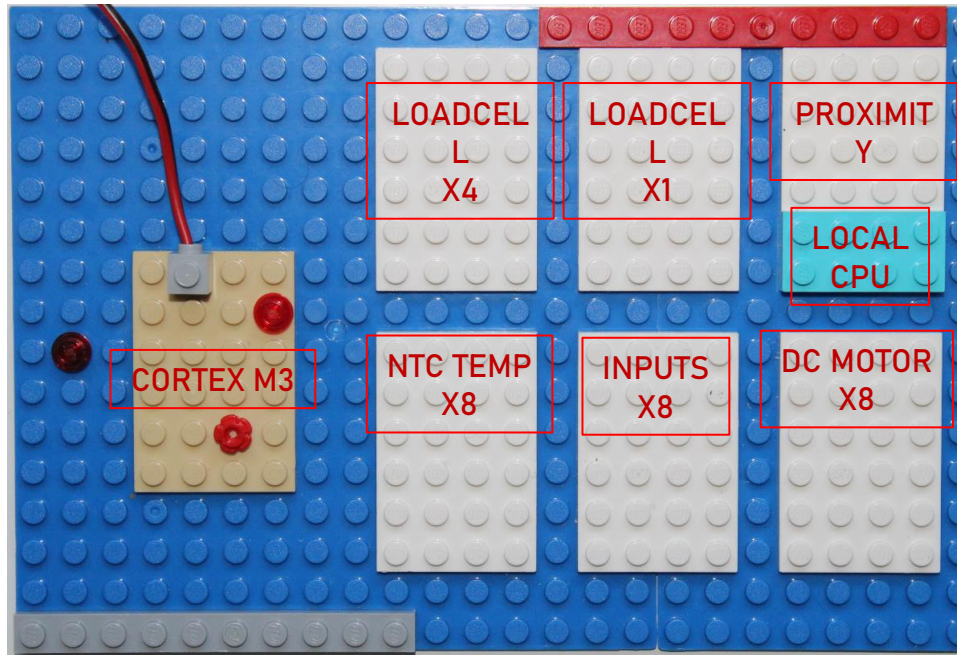


Cloud  
Dashboard  
Getting MQTT  
Data form TTN





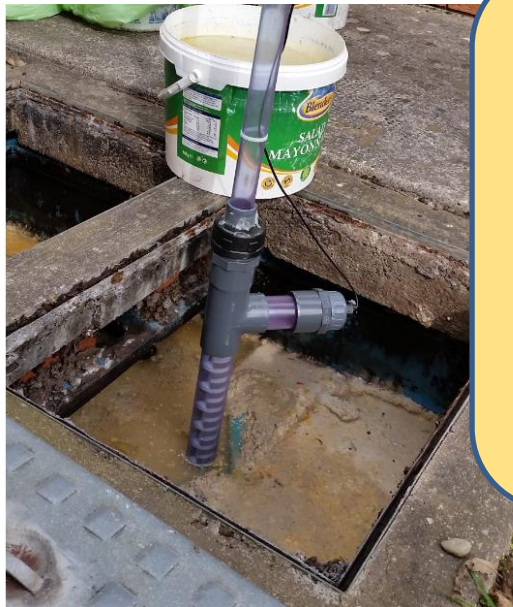
# n-Blocks: Modular Lab Rig





# Use Case – Fats, Oils and Grease Monitor

- Blockages of waste pipes big problem in urban areas
- Monitoring system can help to detect blockages early
- Preventative maintenance of pipes



**Sensors:**  
Custom Capacitive  
IR light,  
RGB,  
Temperature

**Coms:**  
3G, WiFi, Ethernet

# Use Case – Wireless Building Management

- Deployment of wireless sensors in Kalvino library in Torino, Italy
- Prototype development as part of FP7 Tribute project



**Sensors:**  
CO2  
VOC  
Light  
Temperature

**Coms:**  
802.15.4, USB,  
Ethernet

# Use Case – Environmental Monitoring

- Demonstrate deployment in the wild
  - Multiple sensors, processing, wireless communications with backup
  - Energy harvesting



Environmental  
Monitoring

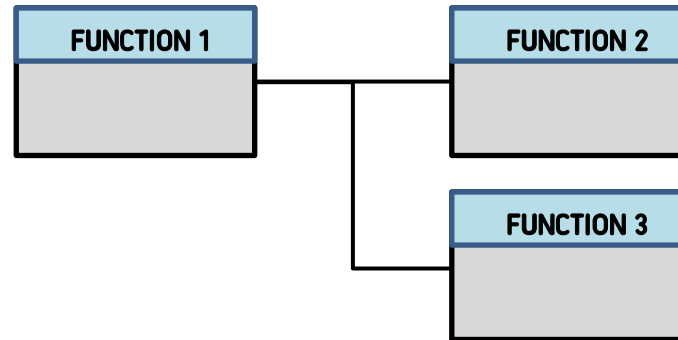


River Liffey  
Deployment

**Sensors:**  
Depth  
Turbidity Water-Temp  
Air Temp  
Humidity Tilt  
Vibrations Light  
Position Flow  
Compass

**Comms:** 3G, LoRa

# nBlocksStudio: Introduction



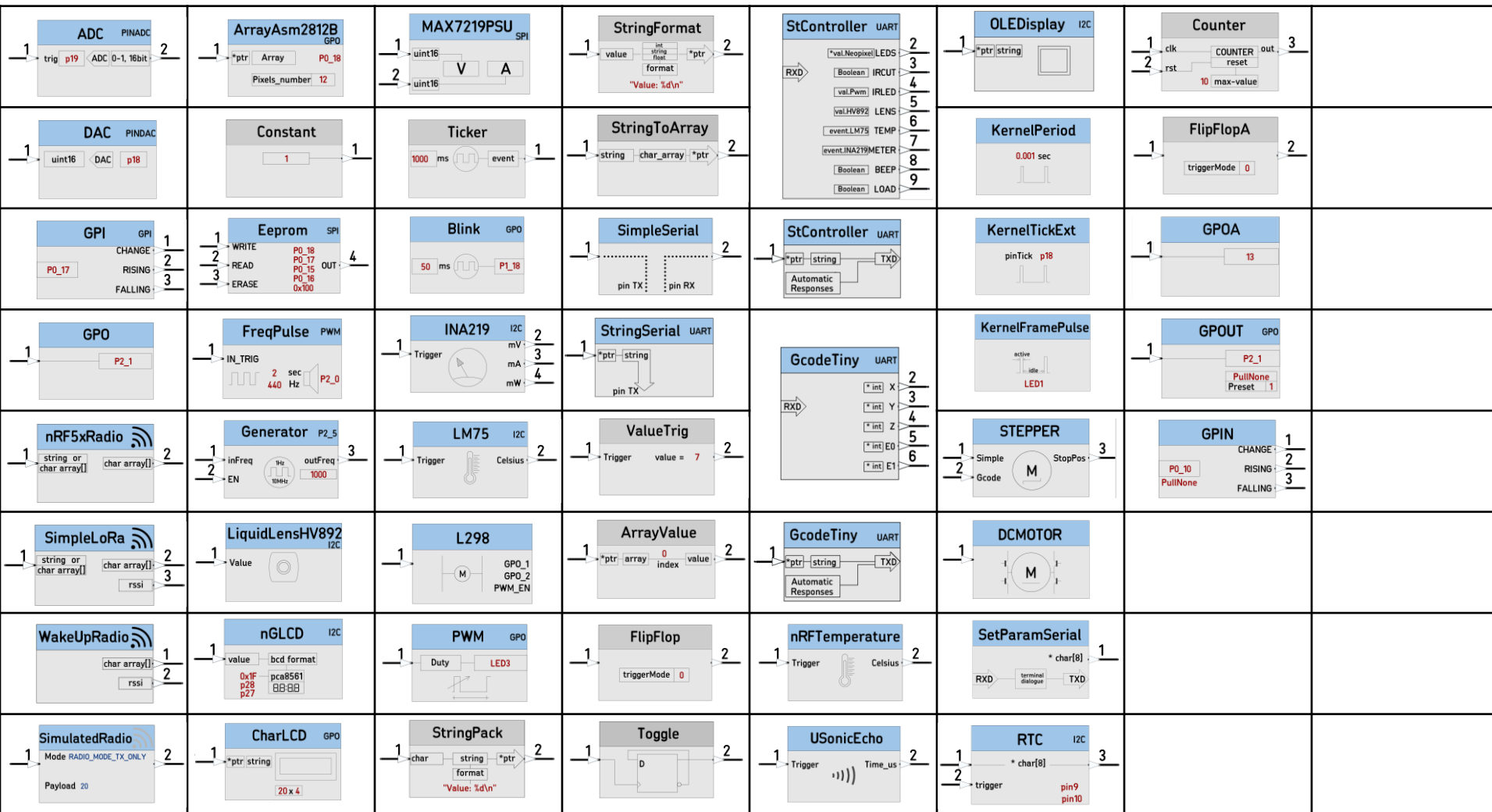
## nBlocksStudio

- Graphical-Diagrammatic programming environment
- Aim – allow users to develop applications without writing code
- Uses the Flow Based Design paradigm
- Function-Nodes connected with Wires

## Code

- Autogenerated code runs in a soft-real-time system
- Underlying layer: Kernel and Event driven tasks
- Contribution server

# Function Blocks



# Concepts



START

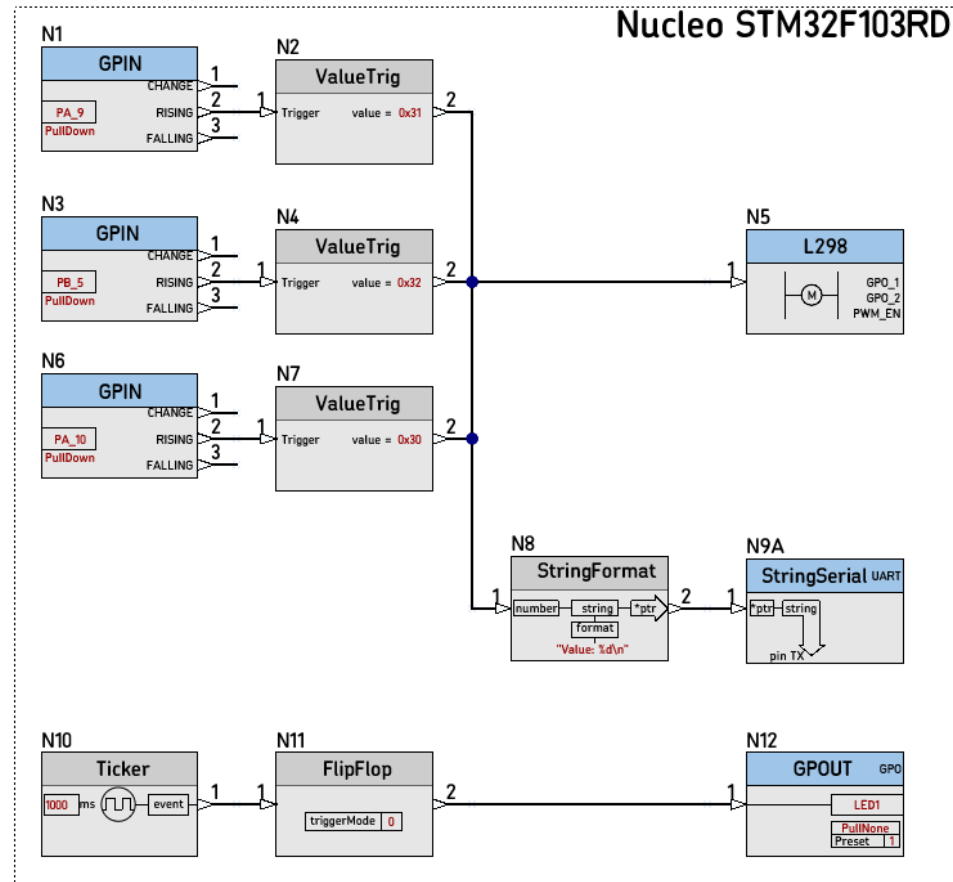


REVERSE

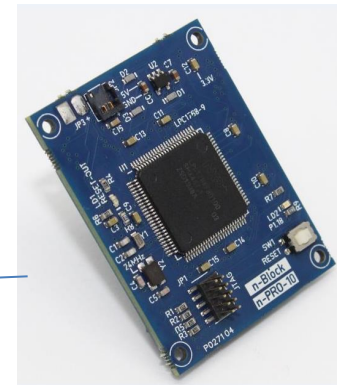
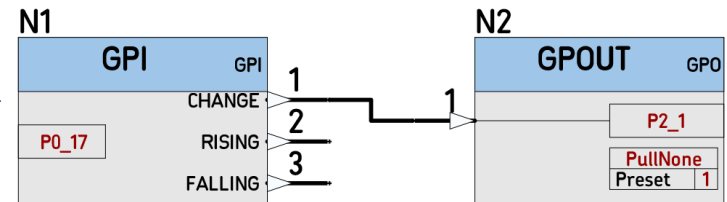
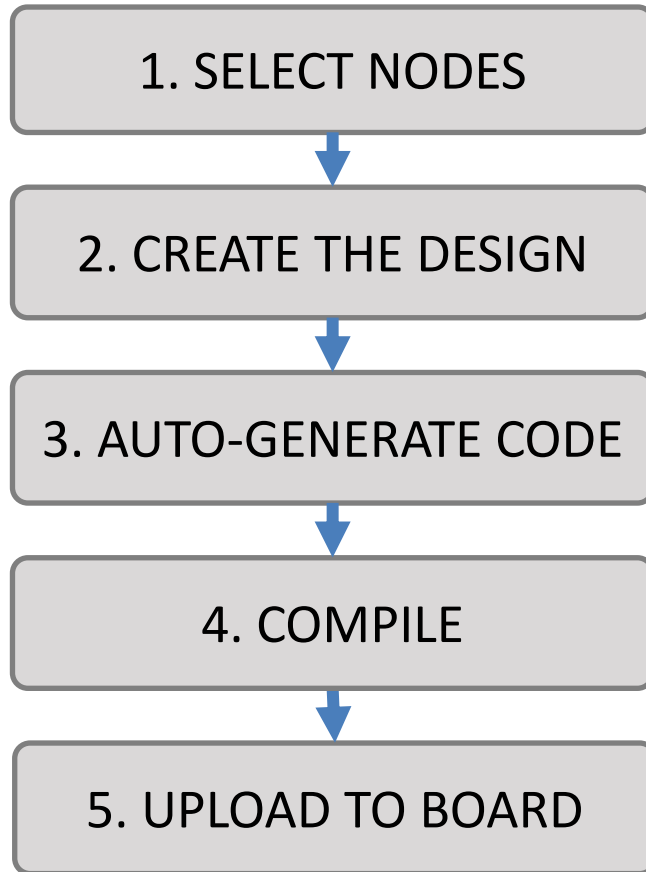


STOP

1. Node
2. Flow
3. Connection
4. Design
5. Translation
6. Compilation

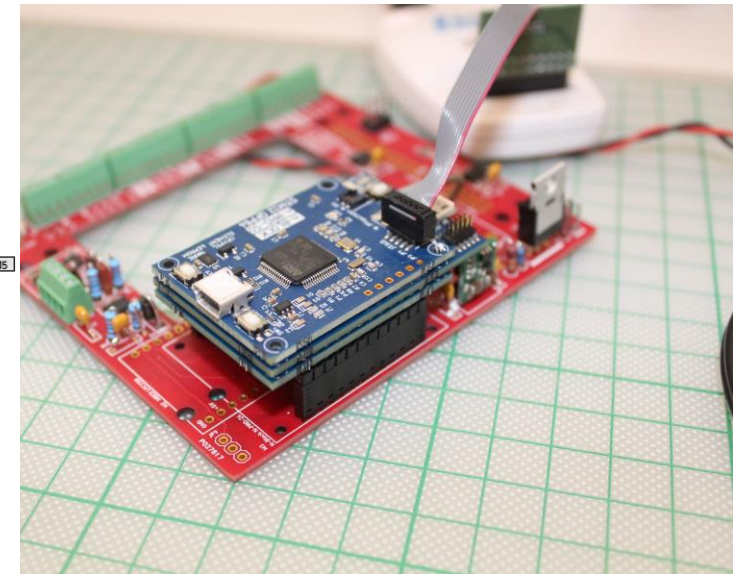
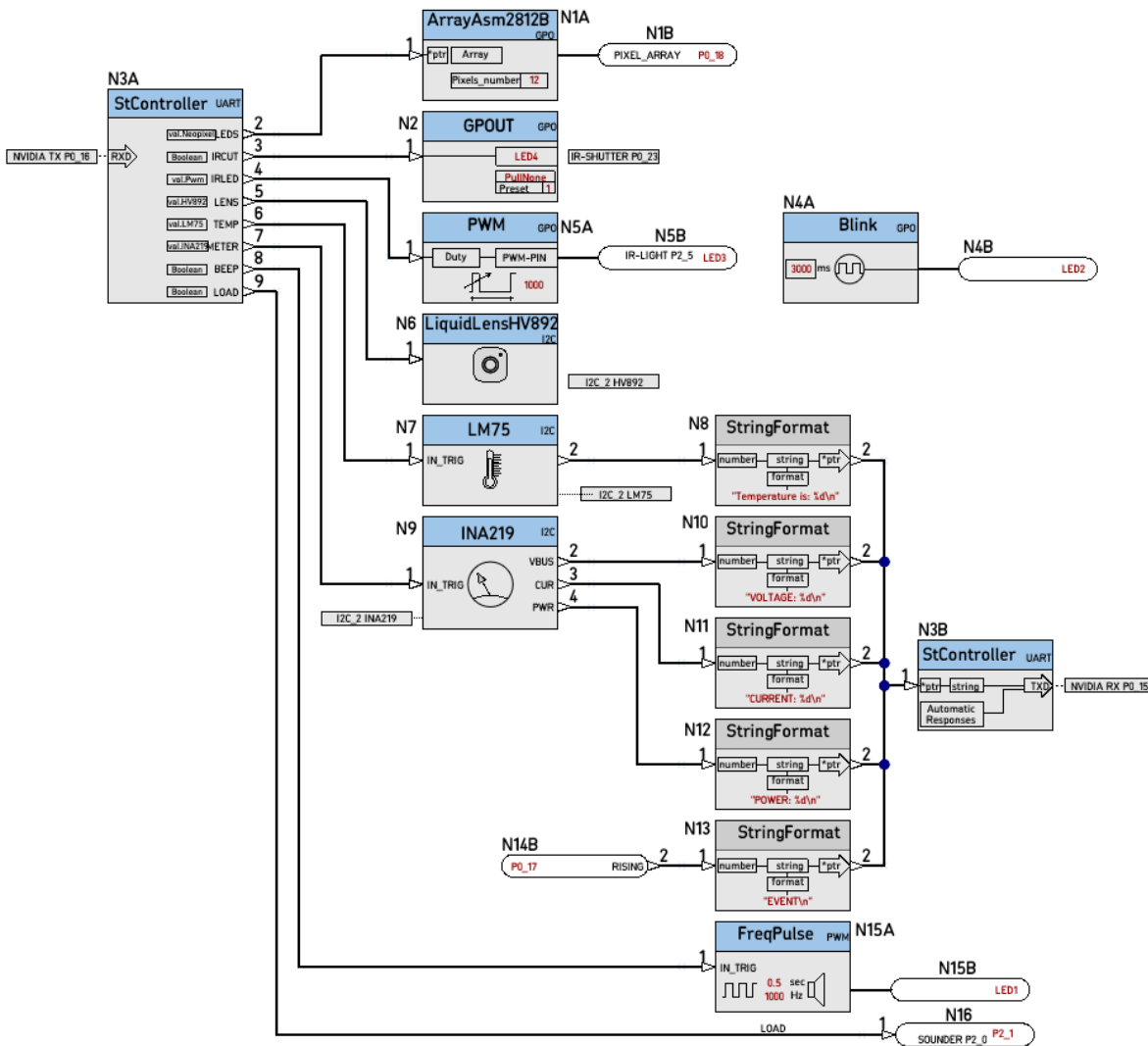


# Workflow



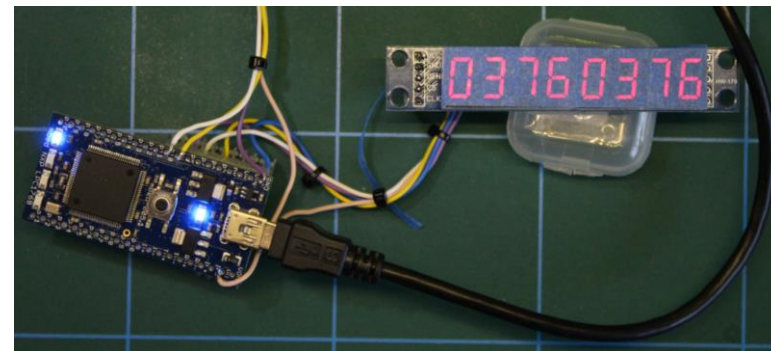
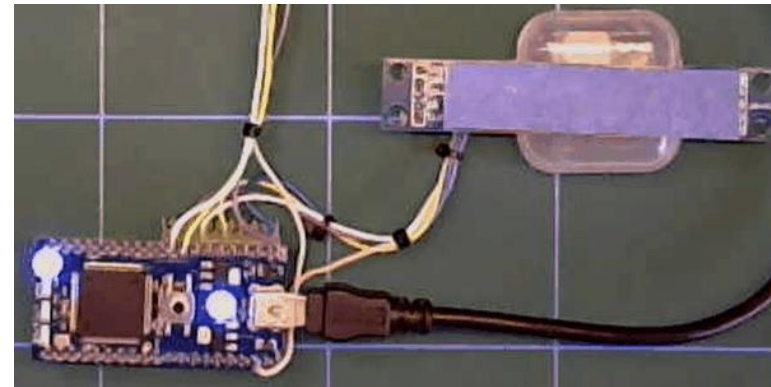
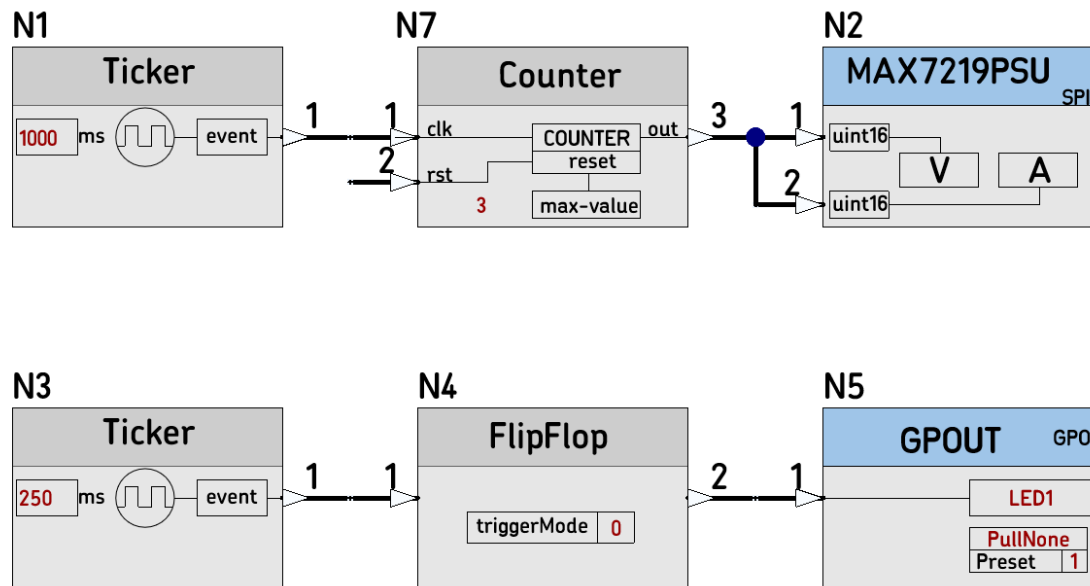


# Example: Optical Scanner Controller

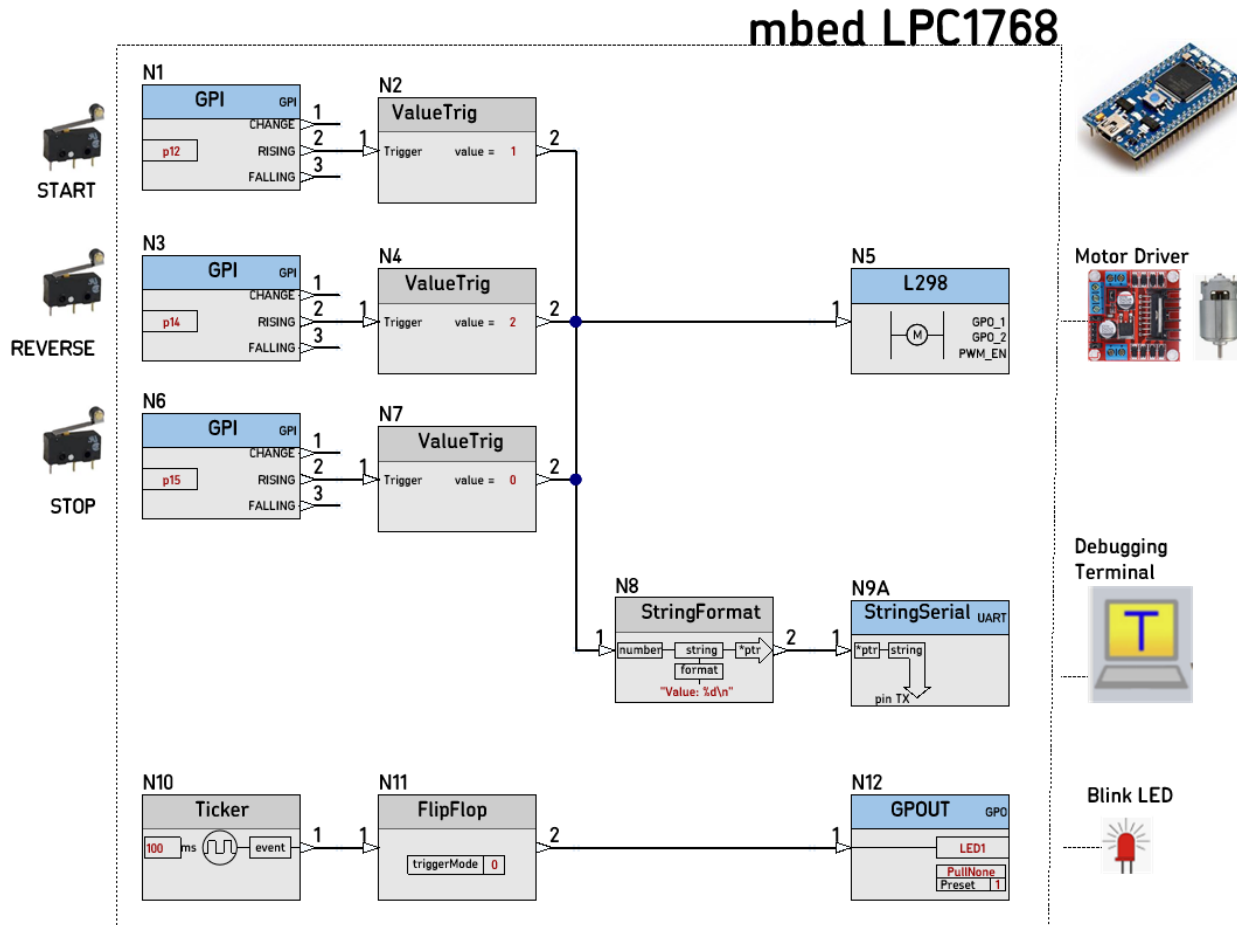




# Example: seven-segment Display



# Example: Simple motion automation



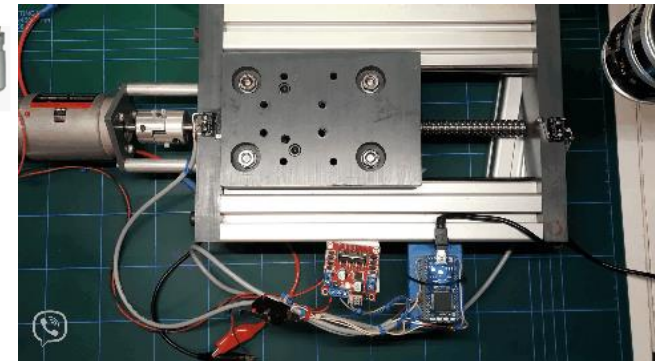
Motor Driver



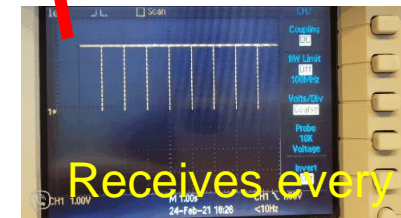
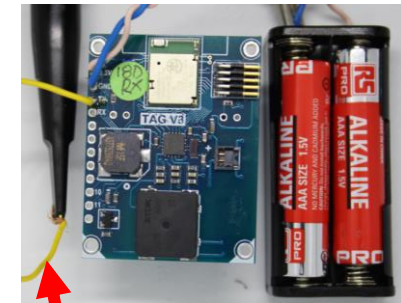
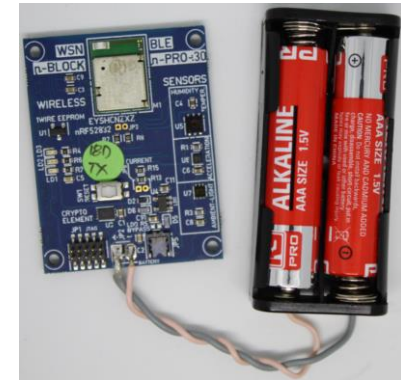
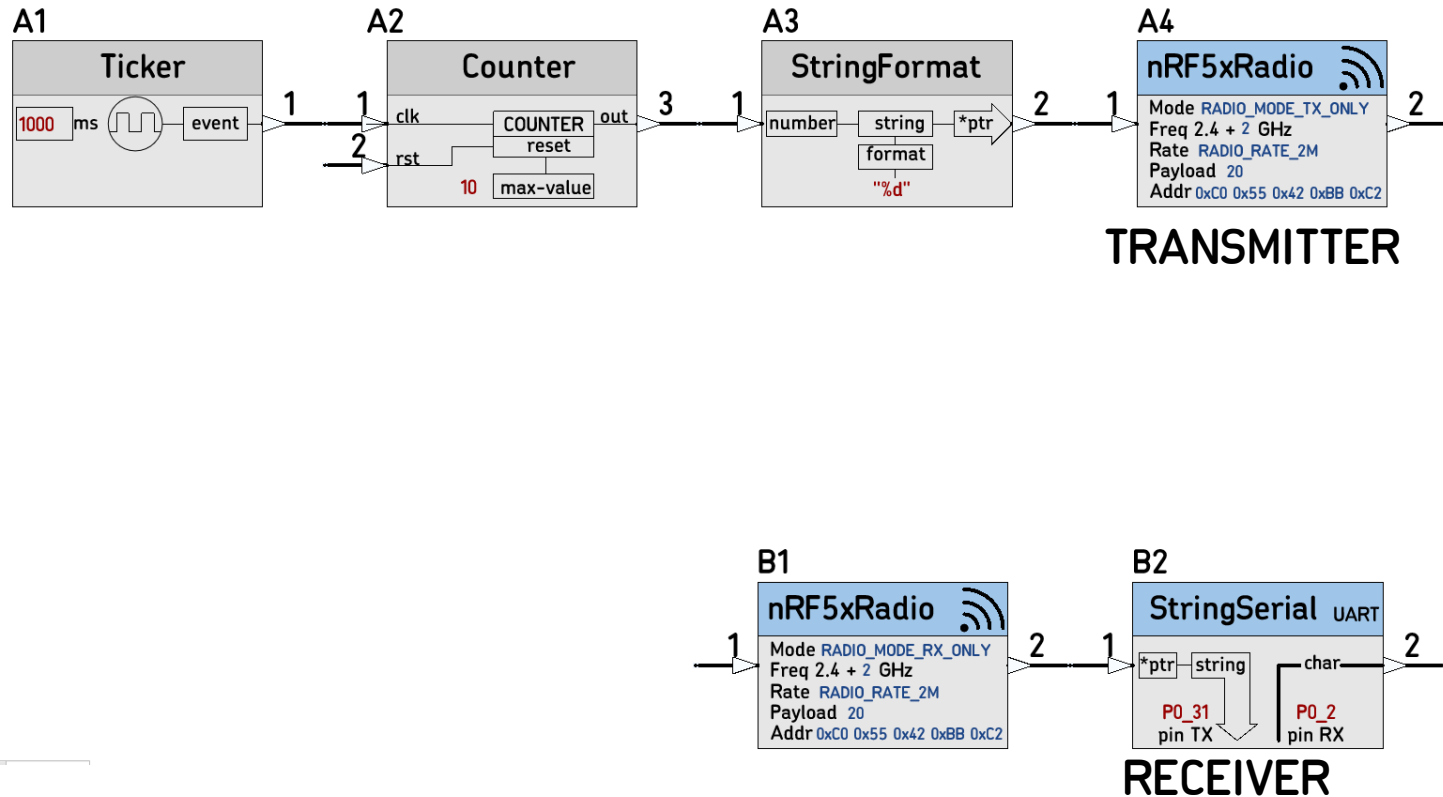
Debugging Terminal



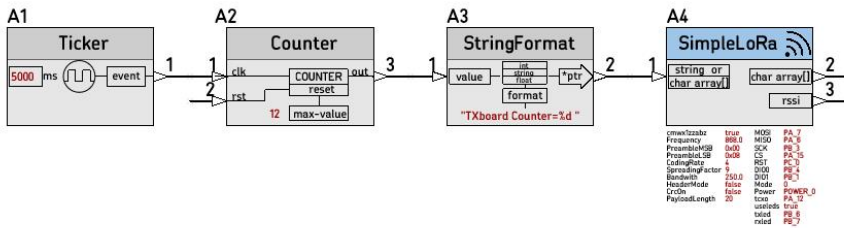
Blink LED



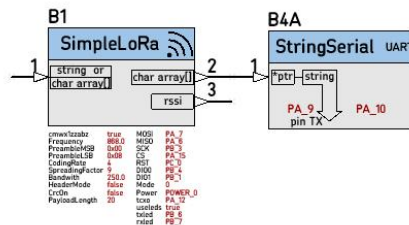
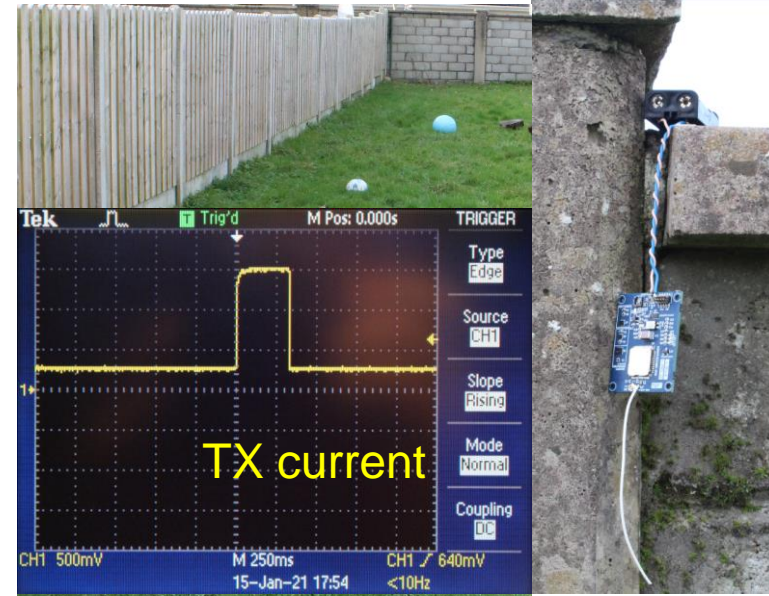
# Example: Simple 2.4GHz Link



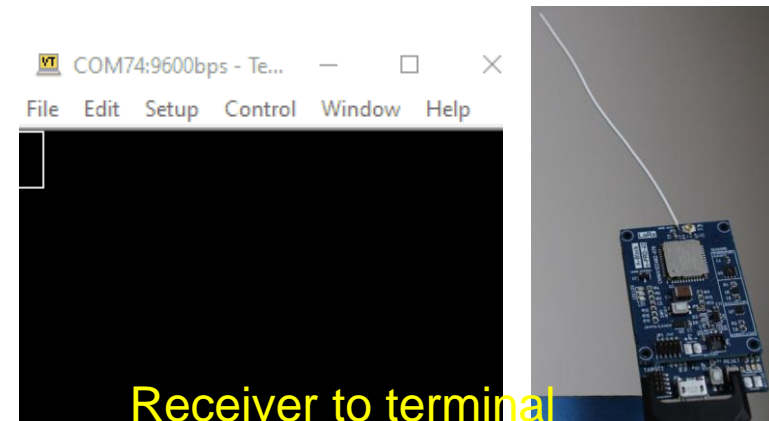
# Example: Simple LoRa Link



TRANSMITTER

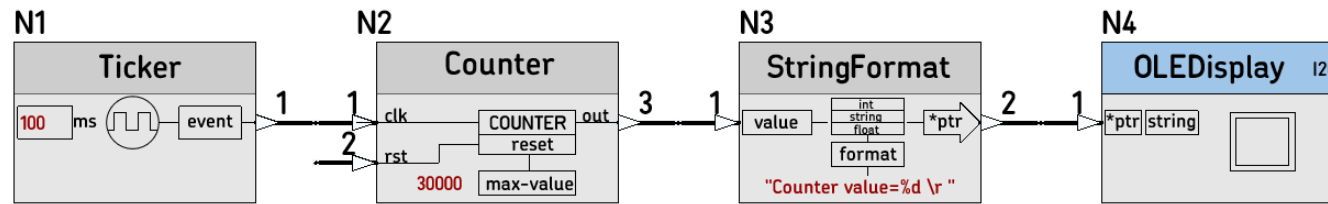


RECEIVER

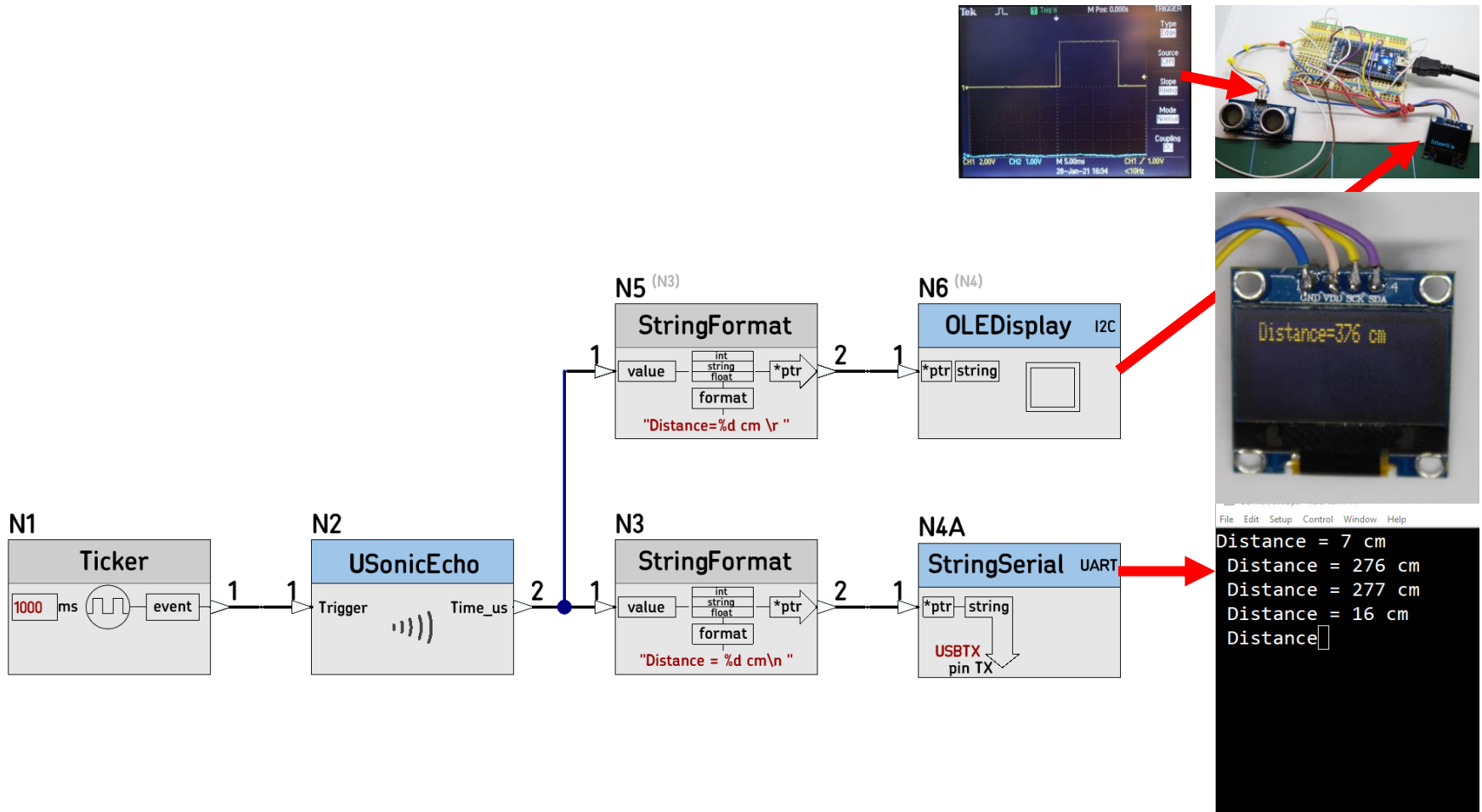


Receiver to terminal

# Example: Counter to OLED display

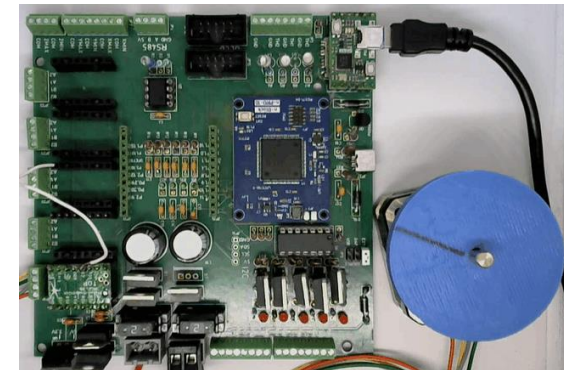
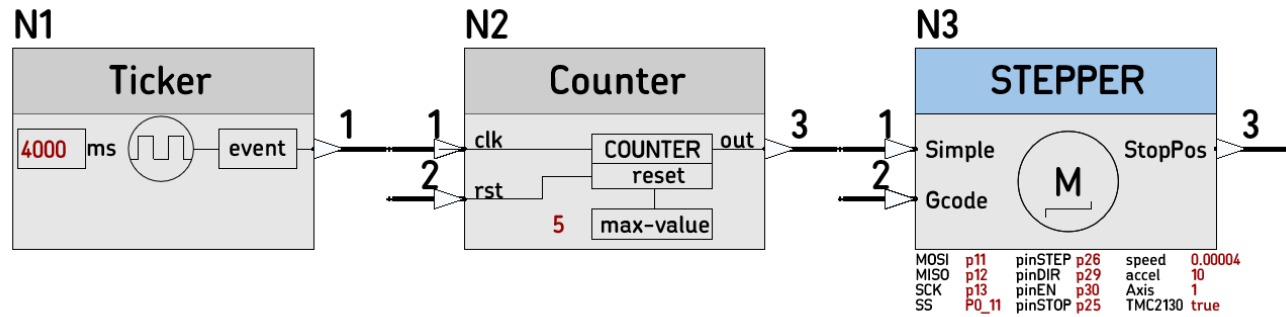


# Example: Ultrasonic Range finder





# Example: Simple Stepping-Motor



# Summary

- **n-Blocks:** A modular low power IoT Hardware platform
  - Uses Standardized Form-factor, Connectors, Interfaces
  - Sensor focused, Powered by ARM Cortex Microcontrollers
  - Applied to real Projects
  - No Cables: Reduces complexity and time to Higher TRL
- **nBlocksStudio:** A flow-based / function-blocks programming environment, for microcontrollers.
  - Uses Function Blocks. Controlled by events and messages
  - Design driven, Powered by C++ classes and abstractions
  - Applied to real Projects and reference designs:
  - No Code: Reduces drastically development time