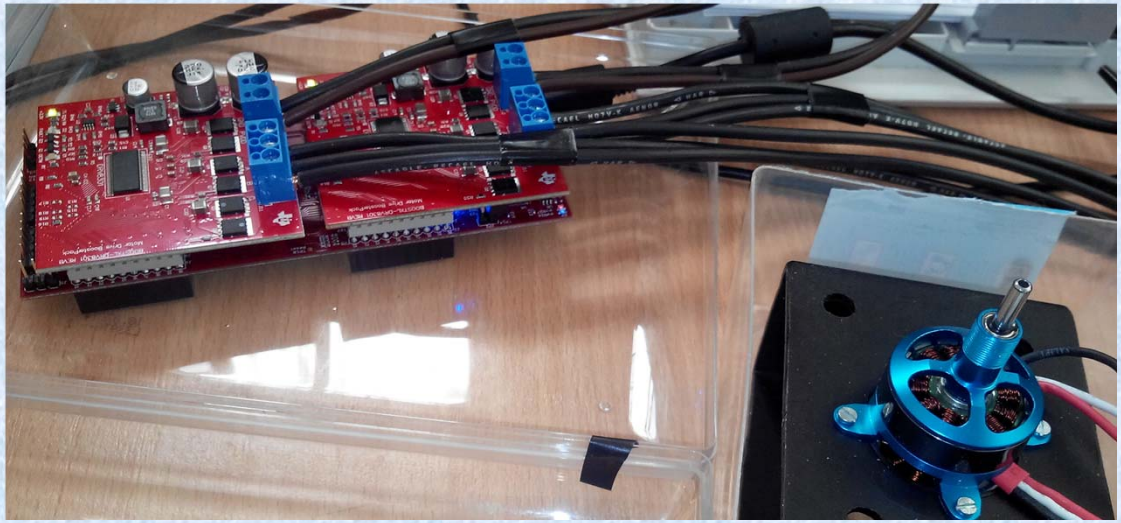
Laboratoire d'Automatique de Tlemcen

Erasmus+

State of the art on µC / DSC for Motor control and Internet of Things



Pr Lotfi BAGHLI
Lotfi.Baghli@univ-tlemcen.dz
Université de Tlemcen

Plan

- Introduction
- What do we need in our domain
- Controllers and communication
- Energy harvesting, energy transfer and IoT
- Electrical devices and Machines
- Conclusion

Introduction

- Need to know:
  - Internet of Things (IoT)
  - Electrical engineering technology
- Applied science, technology
- Simple projects, fast to market

Controllers and communication

- Microcontrollers, DSC
- Interfaces and communications
- Popular boards and shields
- Some applications : data gathering, medical applications, robots

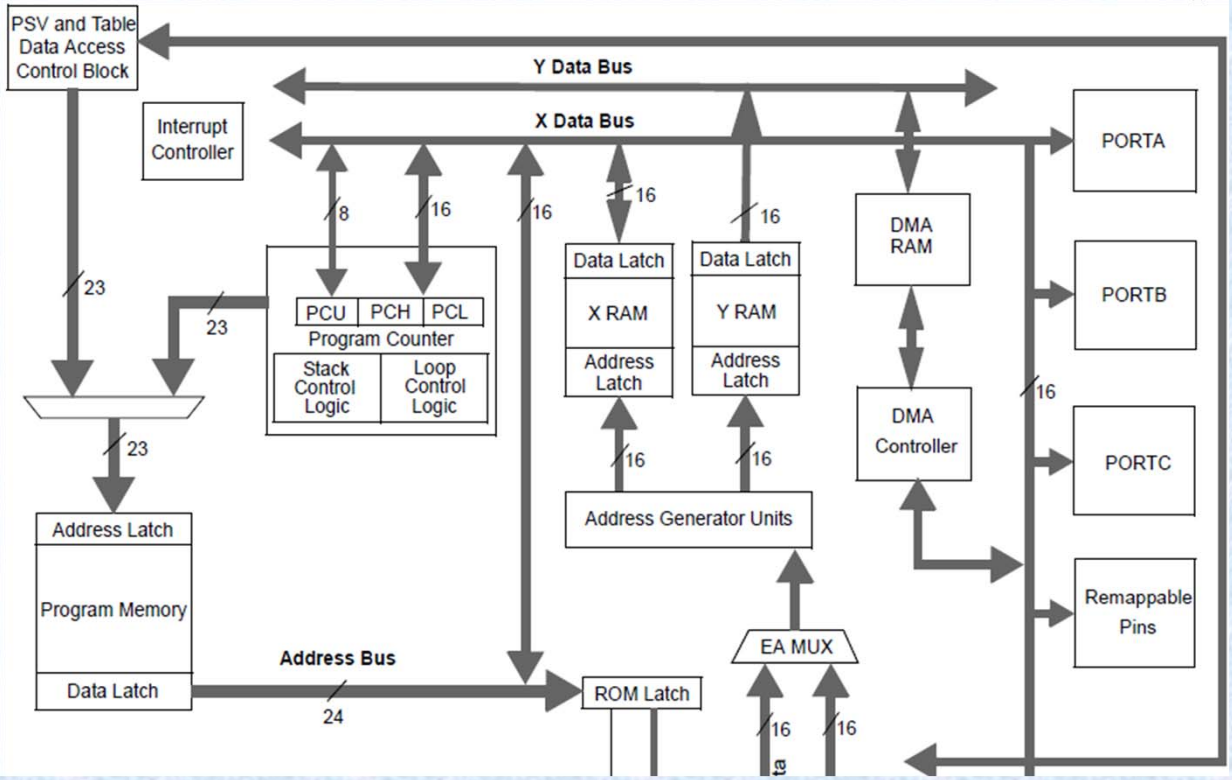
Definitions

- µC : microcontroller = Microprocessor (µP) + memory + peripherals = PC in 1 chip
- DSP : Digital Signal Processor
  - Microprocessor (µP), heart of a computer
  - Additional hardware modules to accelerate some computation operations: Some Of Product (SOP)
- DSC : Digital Signal Controller
  - Microcontroller (µC) + DSP engine in 1 chip
  - Combine the power and computation a DSP with the memory and peripherals of a µC
  - The DSC presents the optimal solution for the real time control of embedded systems that requires high computation.

$$y = \sum_{i=0}^{3} coeff[i] * data[i]$$

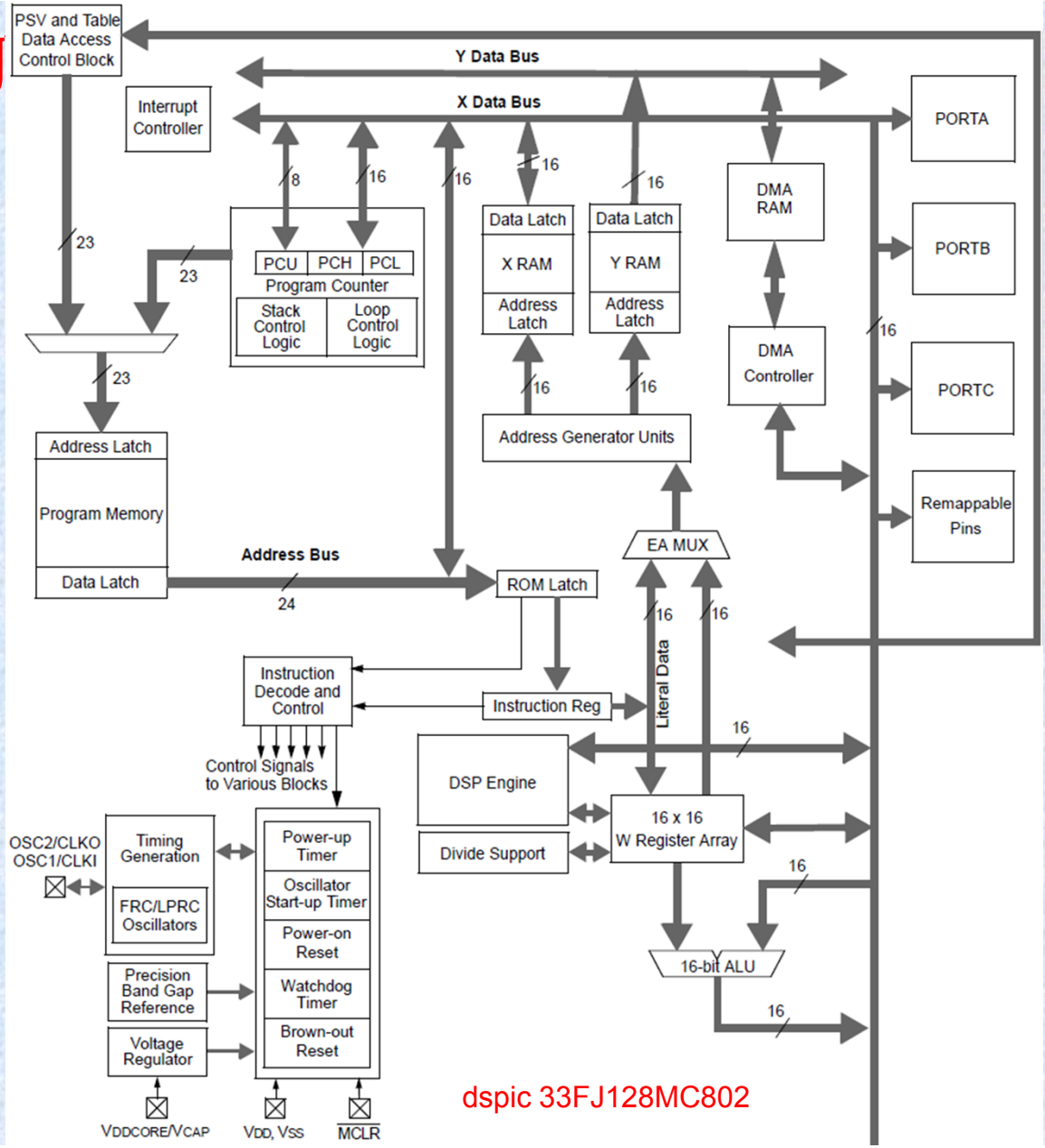
Memory

- FLASH : Program memory, 10k to 100k Erase/Write cycles, Code protected
- RAM : Small but fast
- EEPROM : 5V, 100k to 1M Erase/Write cycles



dspic 33FJ128MC802
3.3V ⇒ no EEPROM

CPU

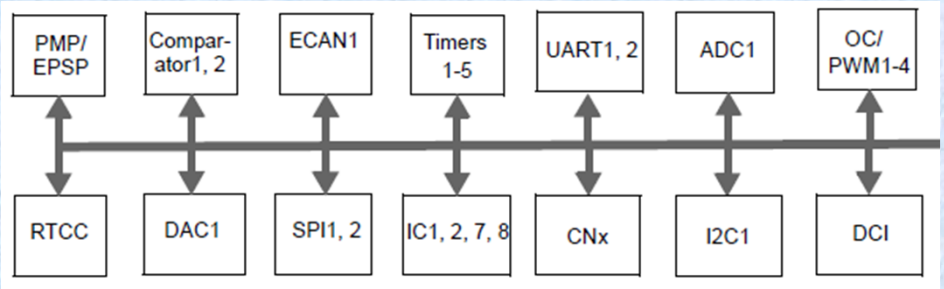


dspic 33FJ128MC802

Peripherals

- Digital Input / Output Lines, pull-up, Change Notification (CN), wake
- Analogue to Digital Converter (ADC), SOC sources, Interrupt.
- Digital to Analogue Converter (DAC)
- Timer / Counter units, prescaler, cascadables, WatchDog
- Pulse Width Modulation (PWM), trigger ADC SOC
- Digital Capture Input Lines
- RTCC
- Communication Interface Units:
  - Serial Communication Interface (SCI) - UART
  - Serial Peripheral Interface (SPI)
  - Inter Integrated Circuit (I2C) – Bus
  - Controller Area Network (CAN)

dspic 33FJ128MC802

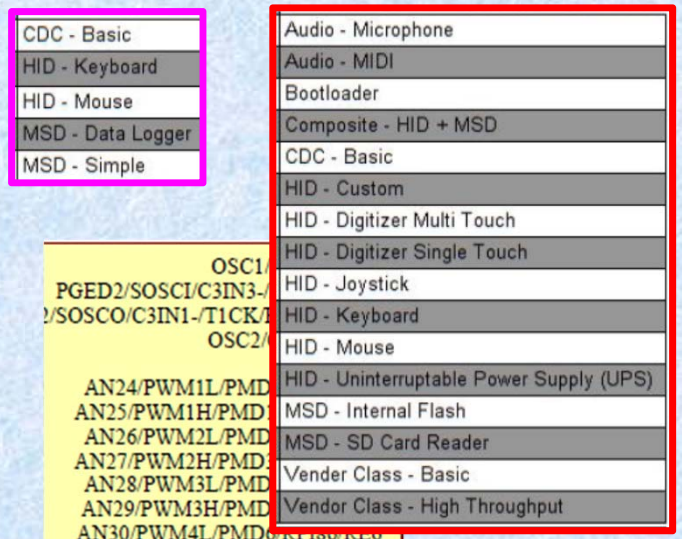


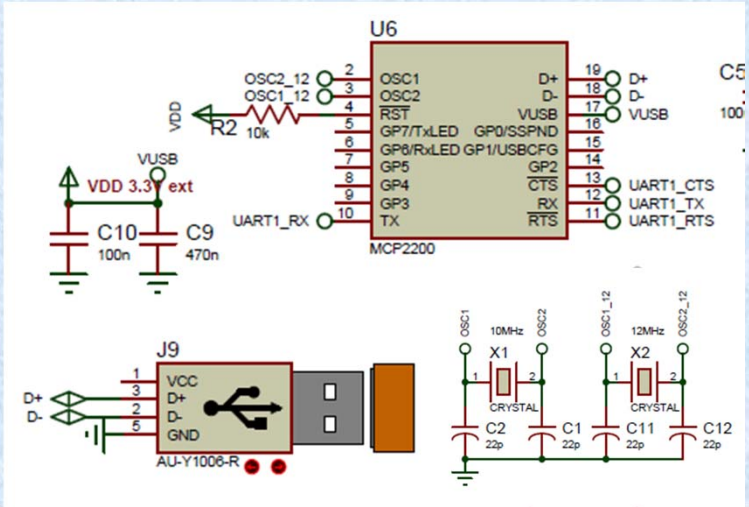
Peripherals

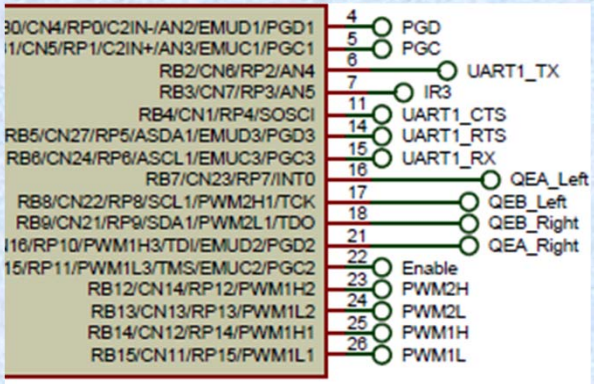
- For IoT communication :
  - Universal Serial Bus (USB)
  - Local / Wide Area Networks (Ethernet)
  - Wifi / Bluetooth (external)
  - Nordic n24L01+ : 2.4 GHz, 1 to 2 Mbps in burst mode, appliances, toys
  - Zigbee
  - Zwave : automation, signal redirection
- Requires software or hardware stack:
  - USB
  - TCP-IP
  - Bluetooth : 3, 4, **BLE**

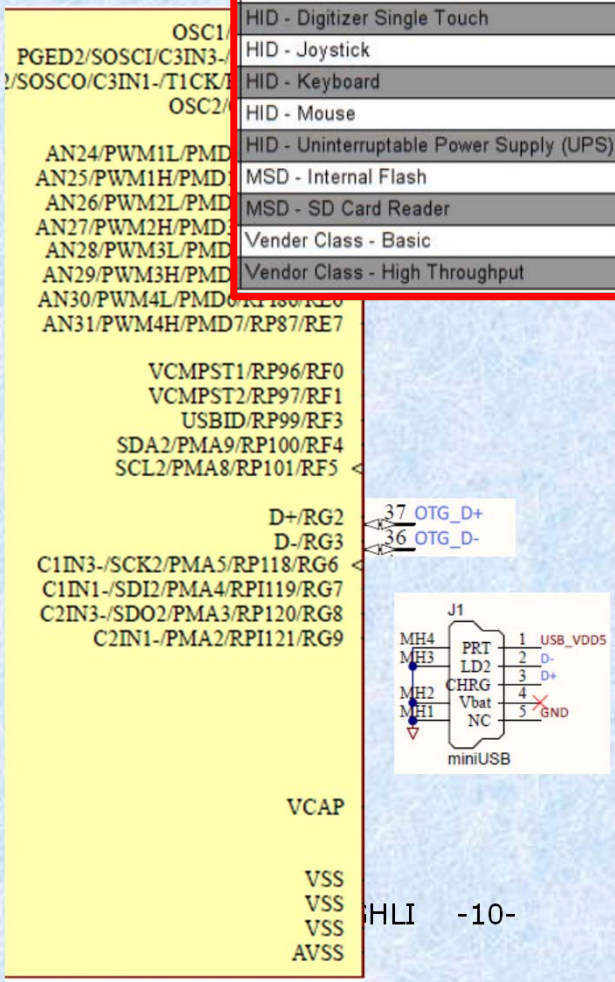
USB

- USB mode :
  - Device mode : CDC, HID (Keyboard, Mouse)…
  - Host mode
  - On-the-Go (OTG)
- Microchip Libraries for Applications (MLA)
- Port COM over USB :
  - FTDI 232RL, 2xxx (Arduino example)
  - MCP2200
  - Ch340, CP210x (Arduino clones example)



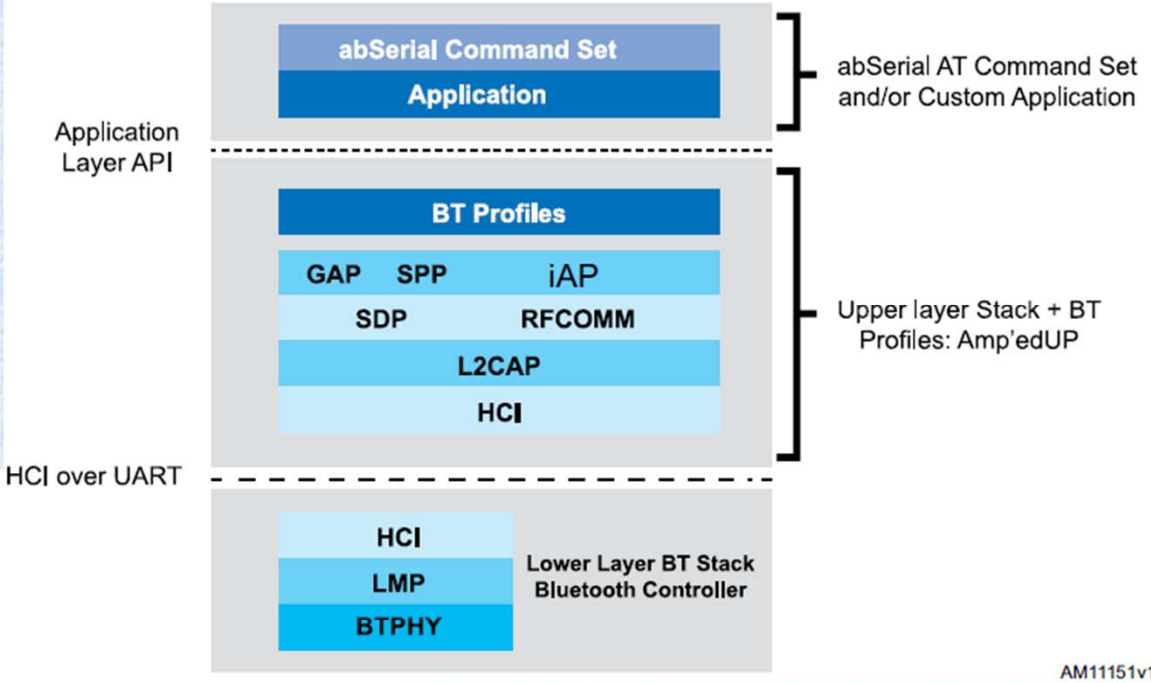


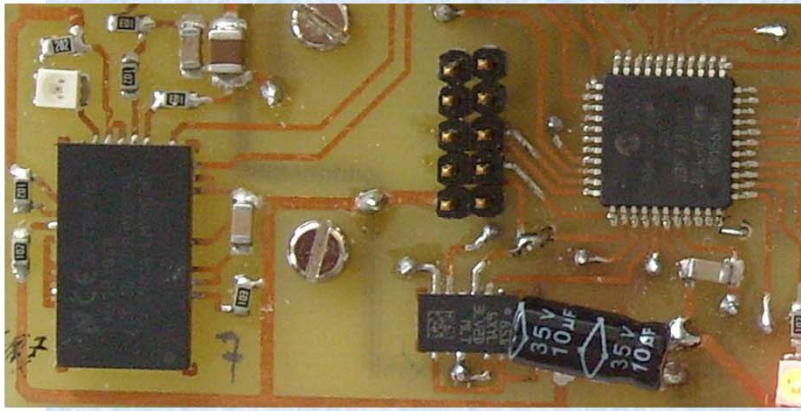




Bluetooth

- Port COM over BT:
  - SPBT2632CA
  - BC-04
  - RN41
  - LMX9838

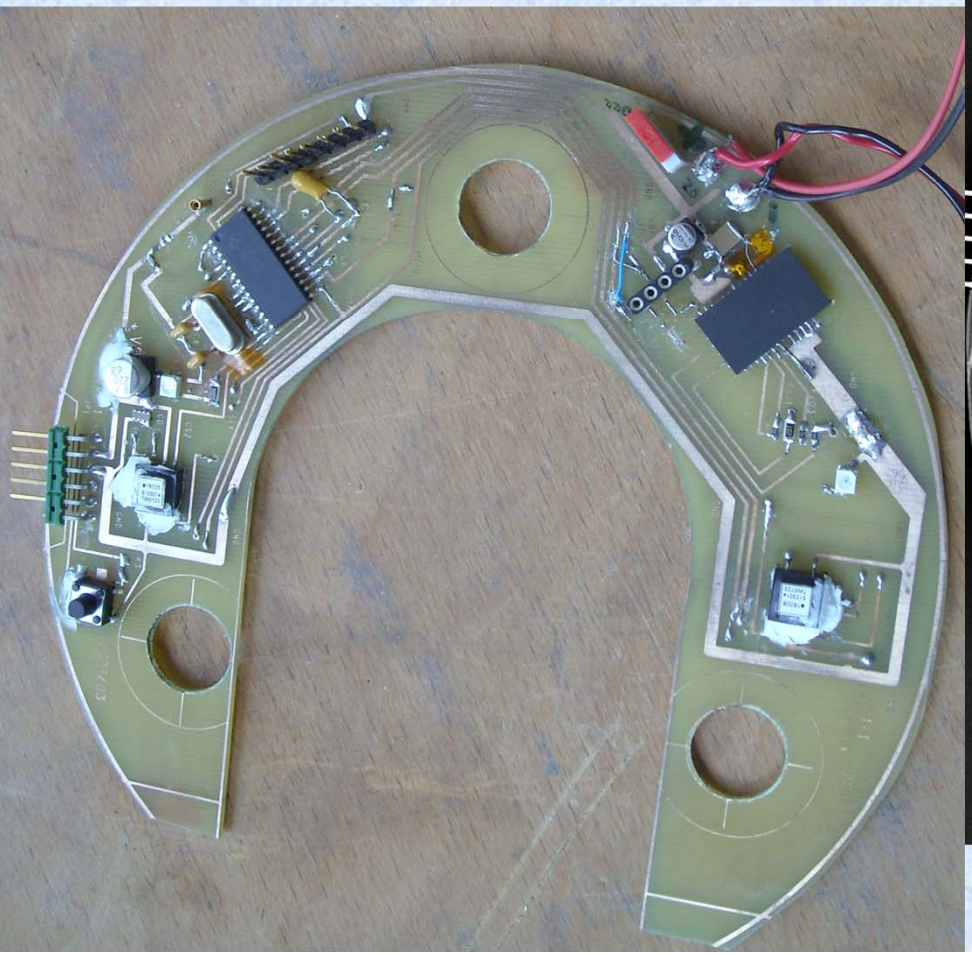


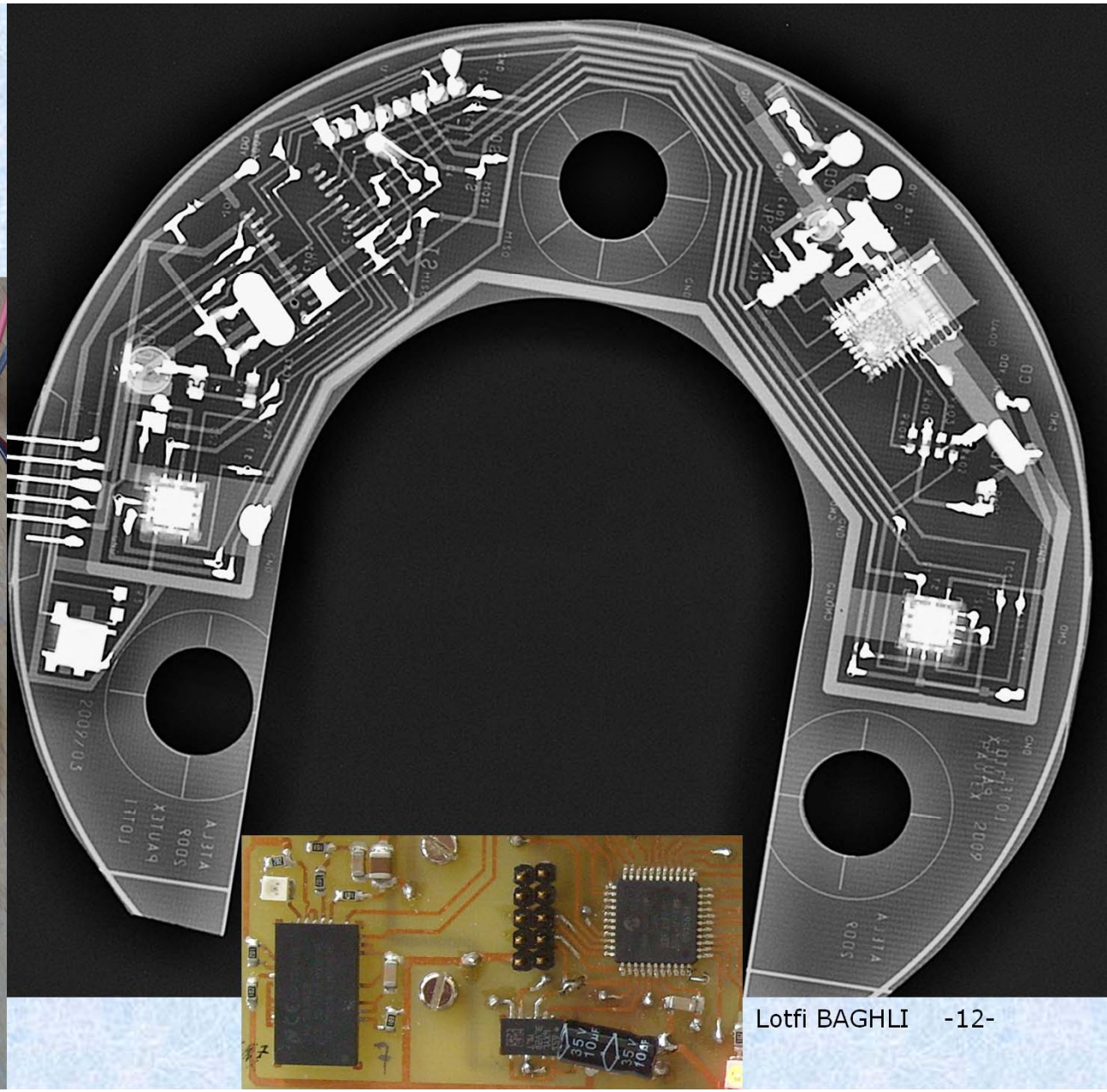


Electrodragon

Bluetooth

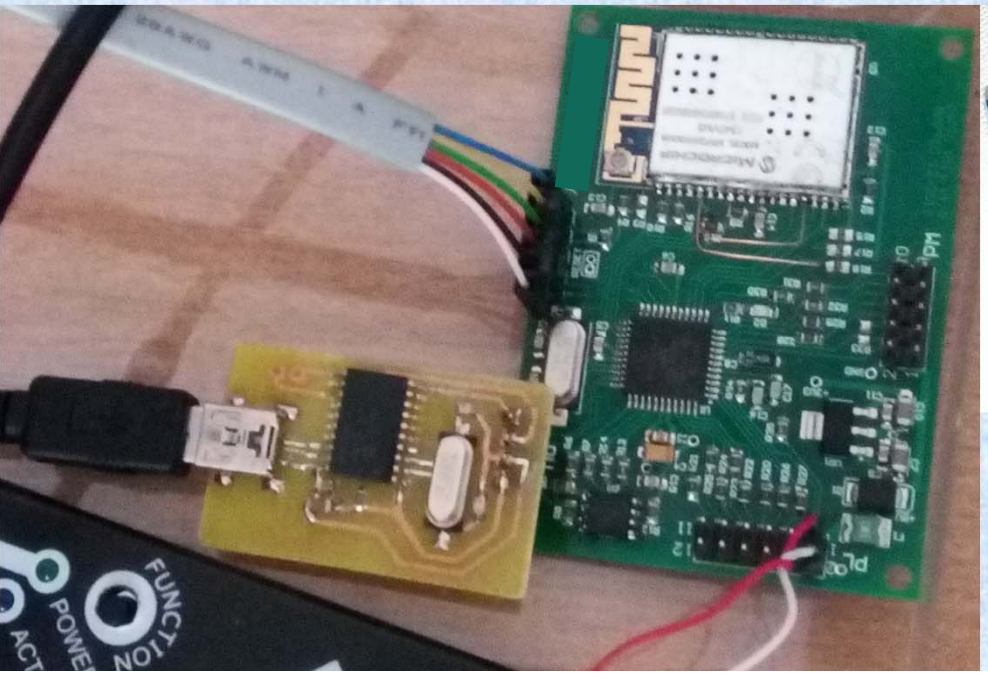
- Port COM over BT:
  - LMX9838

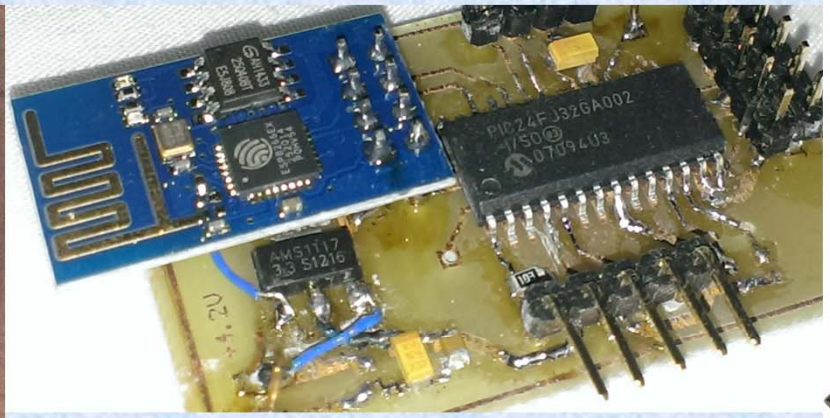


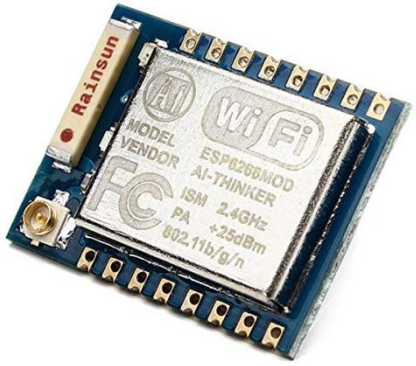


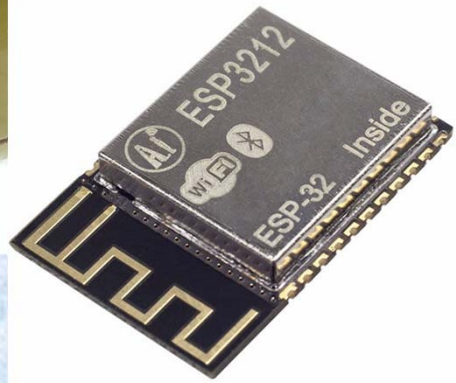
Wifi

- MRF24WG0MA :
  - SPI interface
  - Microchip Libraries for Applications (MLA) : TCP-IP software stack on the host
- ESP 8266 01 / 12 / 14 and a new ESP32
  - UART interface
  - Embedded TCP-IP stack



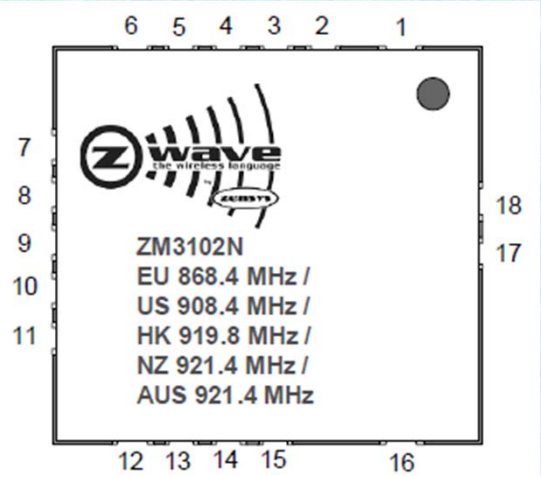


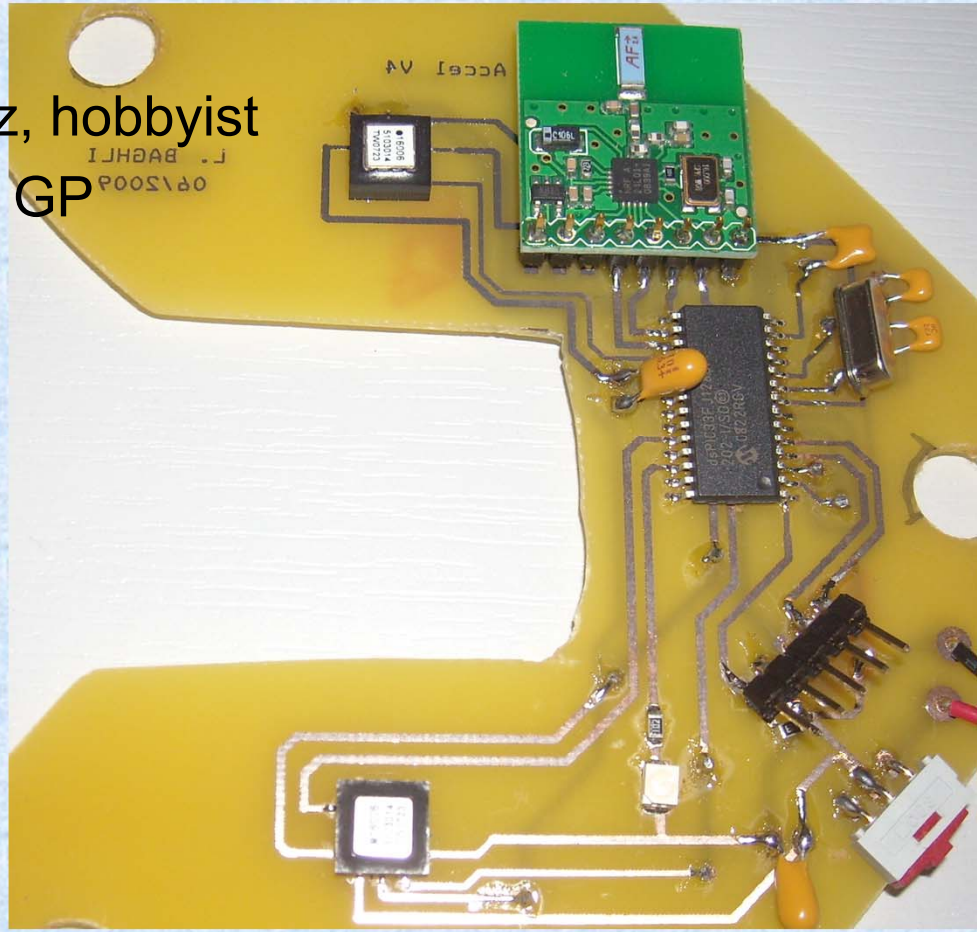




Others

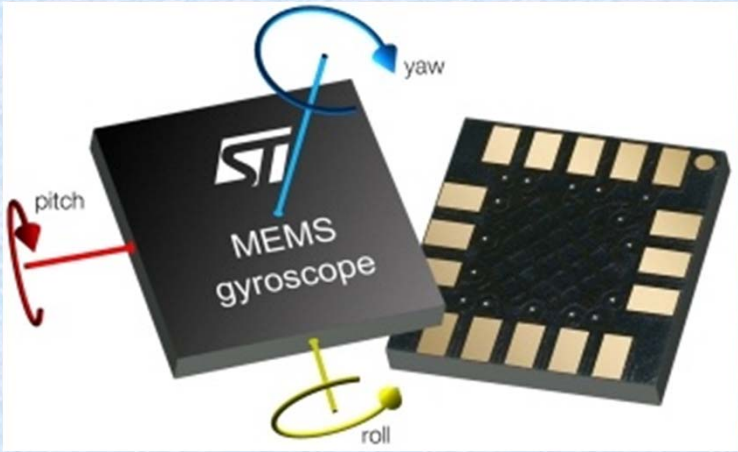
- Nordic n24L01+ : 2.4 GHz, 1 to 2 Mbps in burst mode, appliances, toys, R/C
- Zigbee : 868, 915 MHz, 2.4 GHz, hobbyist
- CC1101 : Multiple frequencies , GP
- Zwave : ZM3102
  - Home automation
  - Multiple frequencies / pays





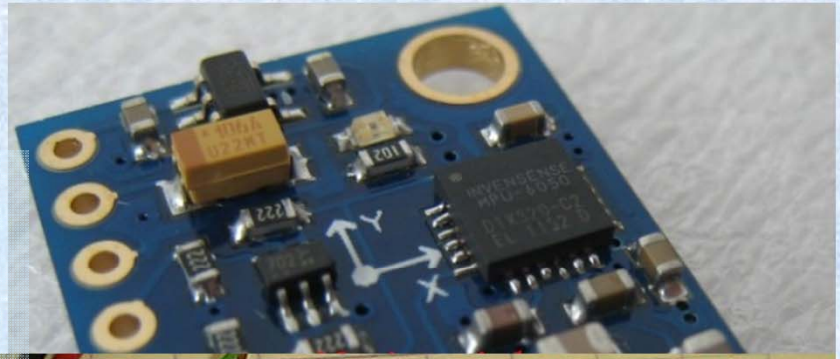
Gyroscope

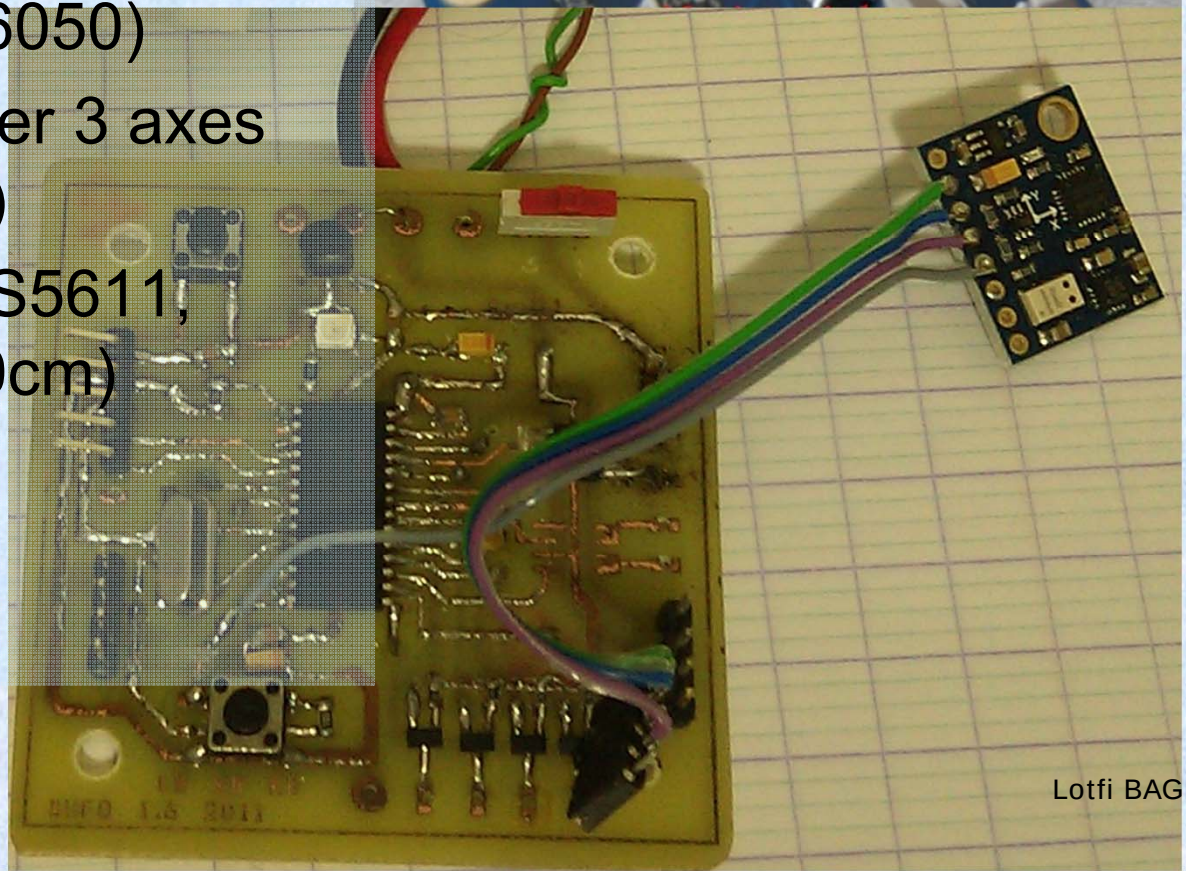
- MEMS Technology
- Analog or Digital output (SPI / I^2C)
- Measure angular acceleration on 1, 2 or 3 axes
- Embedded applications (Quadrirotor, Segway…)



IMU

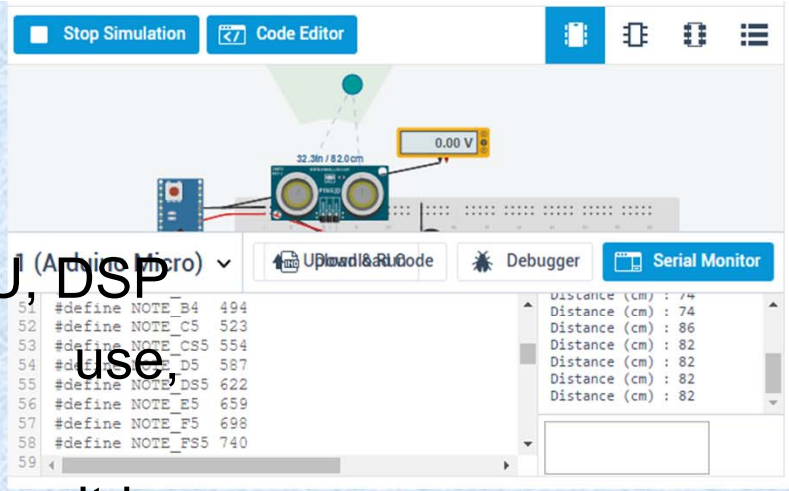
- MEMS Technology
- Accelerometer + Gyro 3 axes (MPU-6050)
- Magnetometer 3 axes (HMC5883L)
- Altimeter (MS5611, resolution 10cm)

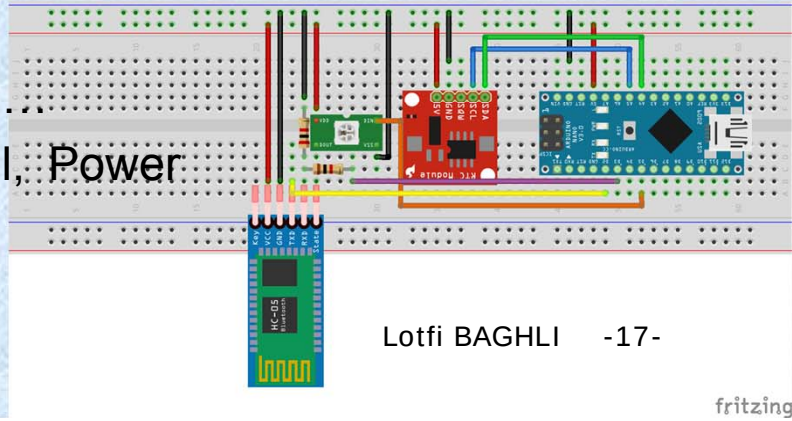




Choice criteria

- Test board and components prices
- Performances : 8/16/32/64 bits, MHz, FPU, DSP
- Development tools: free, easy to use, debugging, HMI
- Simulation and rapid prototyping tools: circuit.io, Fritzing, Altium, Eagle, Proteus
- Popularity: forums, tutorials, user's guides, application notes
- Usage categories:
  - Initiation to microelectronics (µC)
  - Student projects / college: robots, interactions,…
  - Master and research in EE: Machine control, Power electronics, communication, IoT

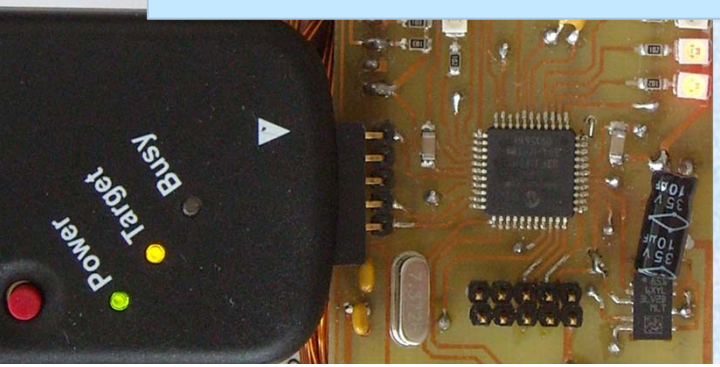


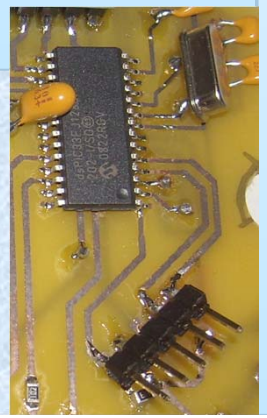


Categories

Componants

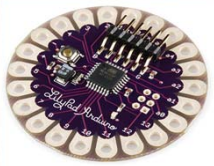
- 8 bits : AVR, PIC 18F, MSP430
- 16 bits : PIC 24F, dspic 30F, 33F, 33E
- 32 bits : PIC 32, Piccolo F28027F / F28069M, Delfino F283xxx, STM32, Cortex M3, M4
- Requires circuit board
- Connector et programer (ICSP, JTAG, ISP AVR : 5 to 500 USD)
- Quartz+2C, PB, LED, Connecters, Pads
- Optimal solution / hit market: Size and cost

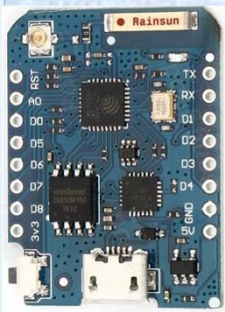




Development boards

- 8 bits : Arduino Uno, nano, Lily pad, LaunchPad MSP430, Mplab Xpress PIC18855
- 16 bits : Curiosity PIC 24F, dspic 30F, 33F, 33E
- 32 bits : Teensy 3.2, Node MCU, ESP8266, ESP32, Wemos, MBed, STM32 nucleo, LaunchePad C2000
- 64 bits : Raspberry Pi3, Pi zero
- Best price: 3 to 45 USD
- Ready to use, rapid prototyping, Boot loader, USB
- Format DIL (Mbed, STM, Arduino nano, Teensy) nucleo, Launch XL, Arduino uno
- Shields : LCD display, OLED, 7-seg., Wifi, BT, DHT, Motor drivers, light, customize

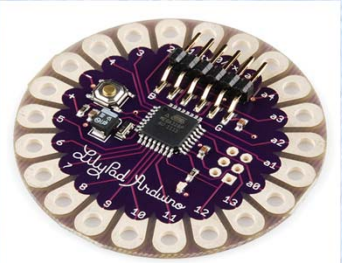




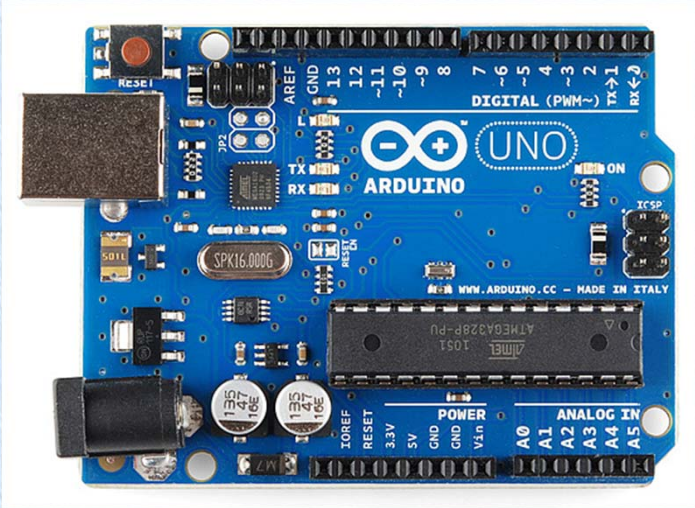
Arduino 8 bits

- UNO : AVR ATmega328P, 5V, 8 bits, 16 MHz, Flash 32 kB (0.5 kB used by boot loader), SRAM 2kB, EEPROM 1 kB
- nano : AVR ATmega328P, 5V
- Lilypad : AVR ATmega328, 2.7-5.5 V
- Shields

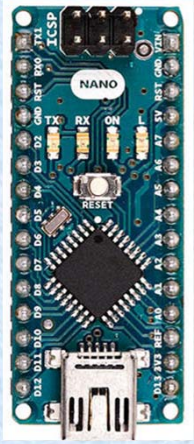
5-20 USD

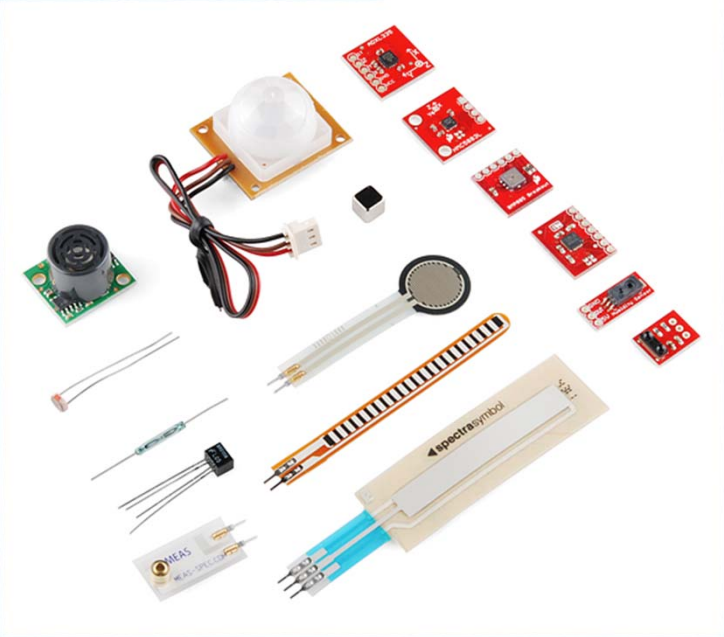


5-20 USD



3 USD



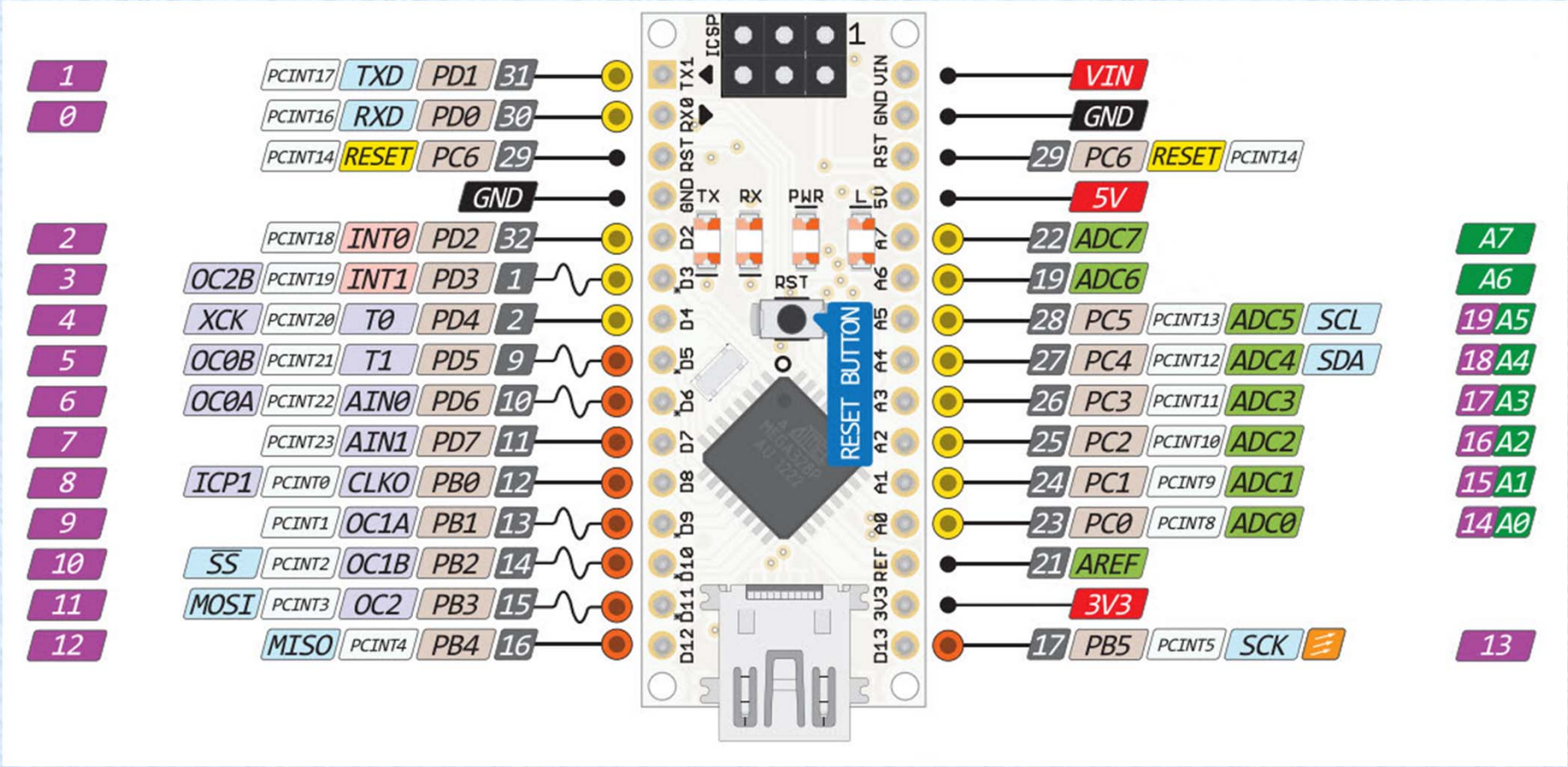


https://learn.sparkfun.com/tutorials/what-is-an-arduino
https://www.arduino.cc/en/Main/ArduinoBoardUno
http://www.arduino.org/products/boards/arduino-nano

Arduino 8 bits

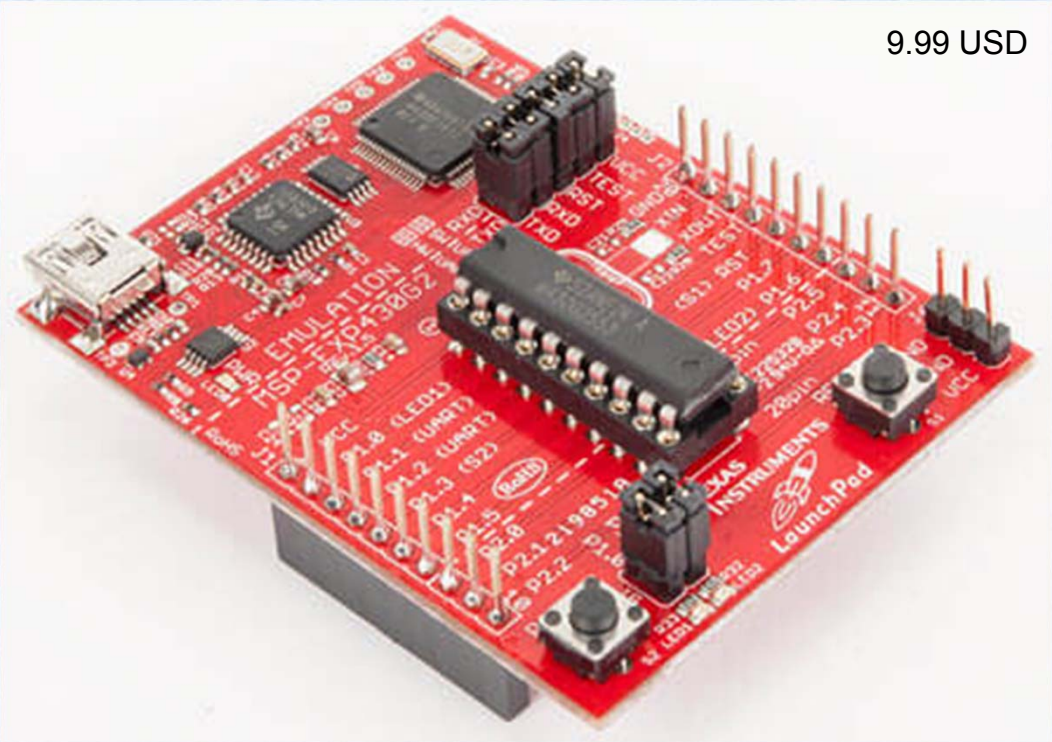
3 USD

- nano : AVR ATmega328P, 5V, 8 bits, 16 MHz, Flash 32 kB (0.5 kB used by bootloader), SRAM 2kB, EEPROM 1 kB



LaunchPad 8 bits

- MSP430 : MSP-EXP430G2 : 16kB Flash, 512B RAM, interruptible GPIOs (capacitive sense-capable), 16-bit timers, 8ch 10-bit ADC, Comparator, Serial Communication (USCI – I2C, SPI & UART) & more
- Courses: MOOC

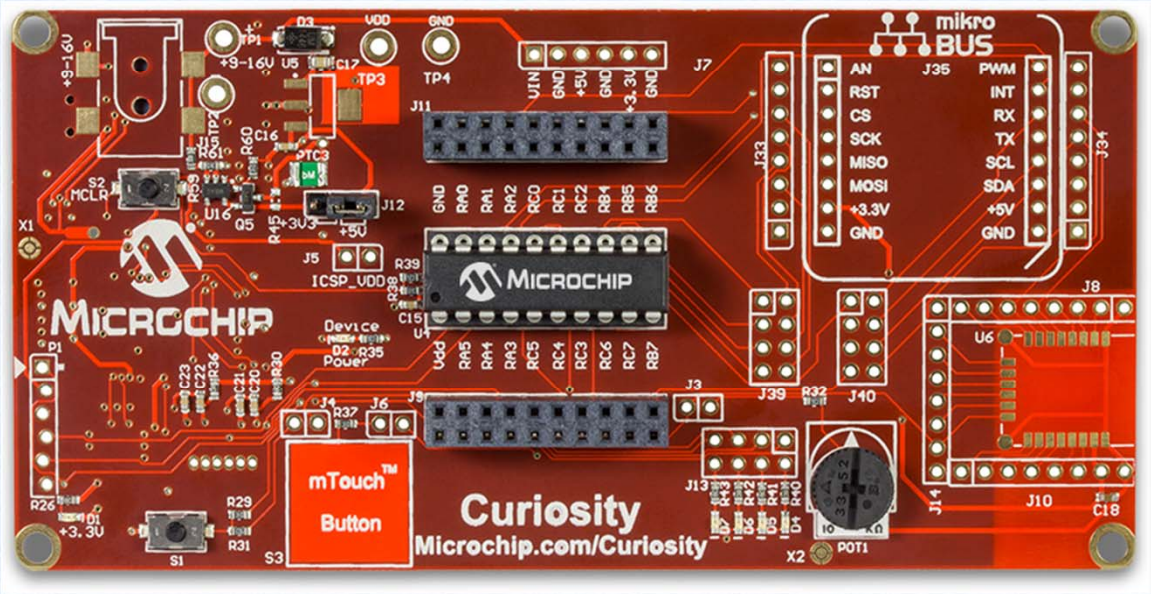


http://www.ti.com/tool/MSP-EXP430G2
https://dev.ti.com/bpchecker/ verify BoosterPack compatibility
http://pyrochat.github.io/mooc-led/cours/106/microcontroleur.html

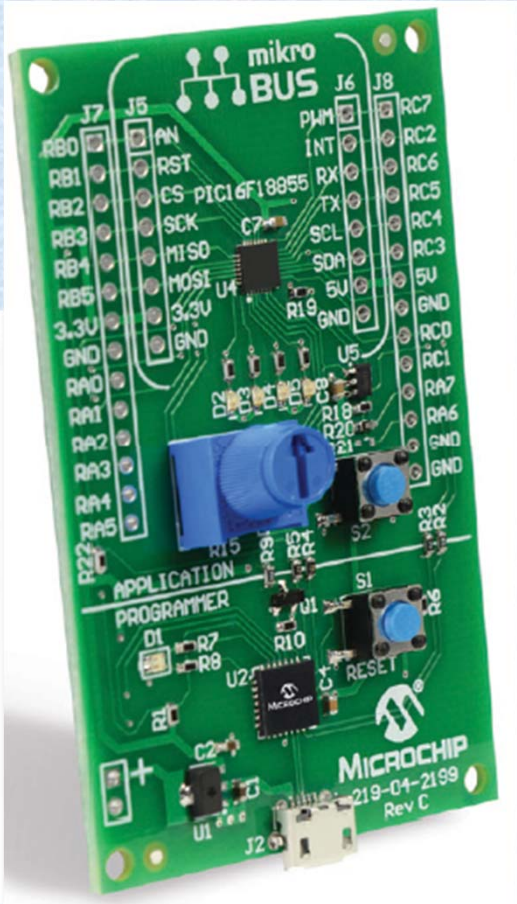
Microchip 8 bits

- MPLAB Xpress : PIC16F18855, 14kB Flash
- MPLAB IDE ou Xpress (cloud based IDE)
- MPLAB Code Configurator
- Curiosity : 8-, 14-, 20-pin 8-bit PIC, Integrated Programmer/Debugger with USB Interface

20.00 USD



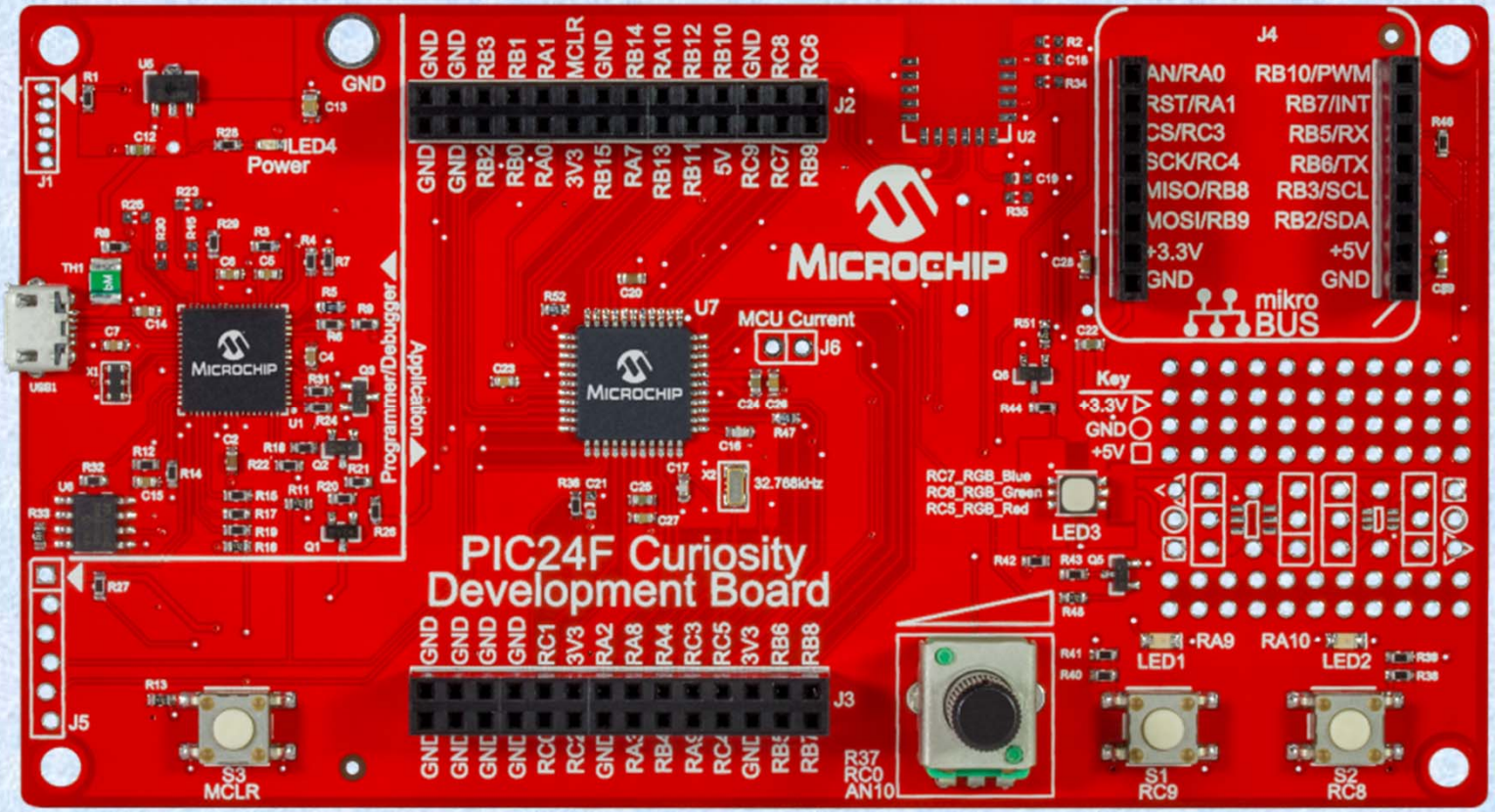
17.00 USD



http://www.microchip.com/mplab/mplab-xpress
http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=DM164137

Microchip 16 bits

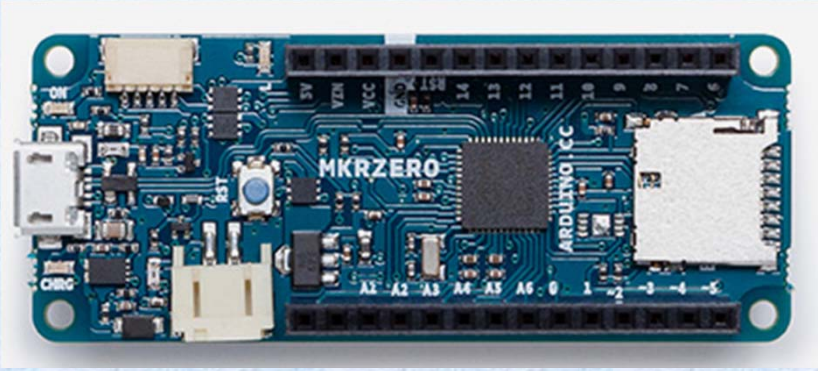
- Curiosity : DM240004, PIC24FJ128GA204, BM71
  Bluetooth LE module footprint 25.00 USD



http://www.microchip.com/Developmenttools/ProductDetails.aspx?PartNO=DM240004

Arduino 32 bits

- MKRZero : SAMD21 Cortex-M0+, 3.3V, 32-bit ARM, 48MHz, 5V in, Flash 256 kB (8 kB used by bootloader), SRAM 32kB, no EEPROM

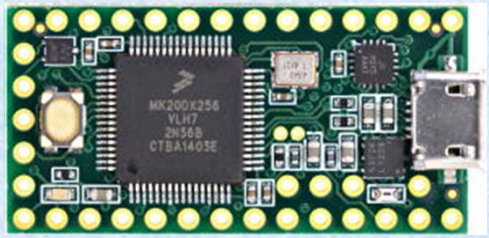


20.90 €

https://www.arduino.cc/en/Main/ArduinoBoardMKRZero

Teensy 32 bits

- Teensy 3.2 : 32 bit ARM **Cortex-M4 72 MHz**, 256K Flash, 64K RAM, 2K EEPROM, 21 Hi-Res ADC, 34 I/O,12 PWM, 7 Timers, USB, 3 UARTs, SPI, I2C, I2S,CAN Bus, IR modulator, I2S, RTC, 16 DMA, Touch Sensor Inputs 20 USD
- Teensy 3.6 : 32 bit ARM **Cortex-M4 180 MHz FPU**, 1M Flash, 256K RAM, 4K EEPROM, uC MK66FX1M0VMD18, 25 Hi-Res ADC, 2 DAC, 62 I/O, 22 PWM, 14 Timers, Hi speed USB, 6 UARTs, 3 SPI, 4 I2C, I2S, 2 CAN Bus, Ethernet mac, microSD, IR modulator, I2S, RTC, 32 DMA, 11 Touch Sensor Inputs, CRC 34 USD



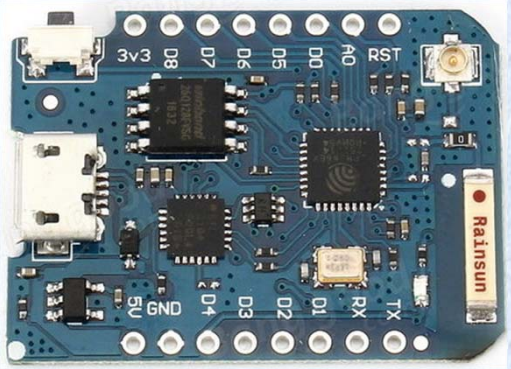
Teensy 3.2

Teensy 3.6

https://www.sparkfun.com/products/13736
https://www.sparkfun.com/products/14058
https://www.pjrc.com/teensy/index.html

Wemos

- Wemos D1 mini Pro : ESP8266Ex, **Wifi**, antenne pcb + FLU, 11 I/O, 1 Analog (3.3V), Interrupt/pwm/I2C/one-wire,16 MB Flash
- Shields: Relay, barometer, DHT, RGB LED, microSD, DS18820



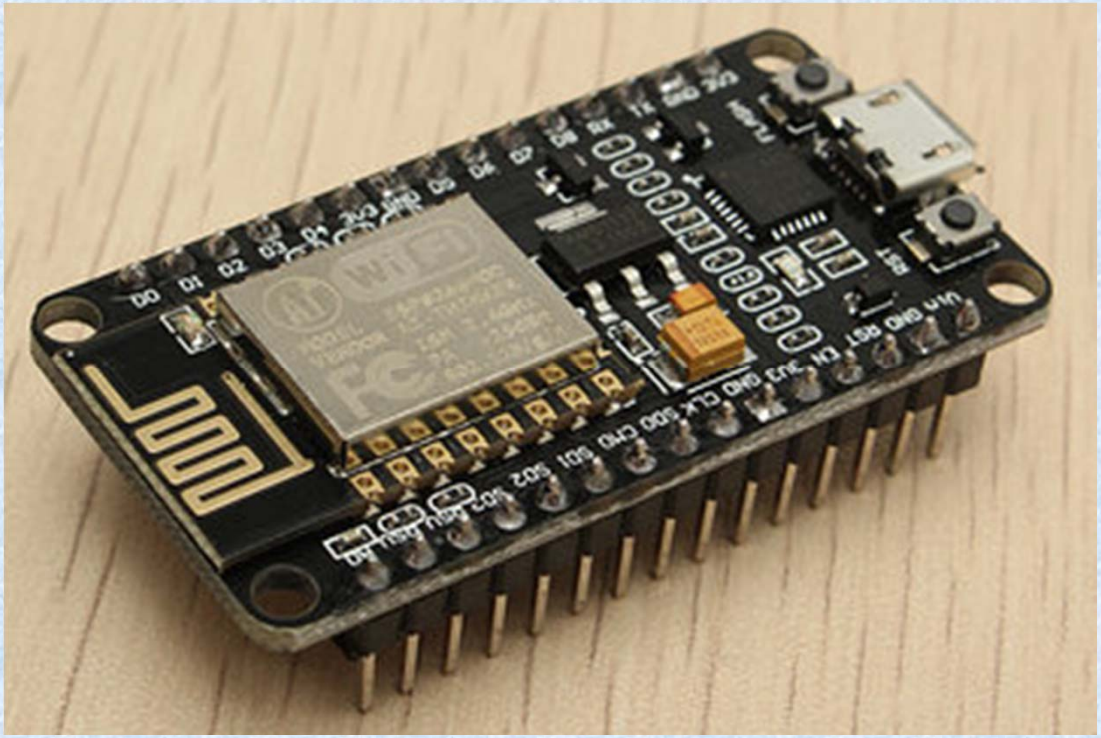
7.77 USD

17.78 USD

http://www.banggood.com/Wemos-D1-Mini-Pro-16M-Bytes-External-Antenna-Connector-p-1107484.html

NodeMCU

NodeMcu Lua WIFI Internet of Things development board based on ESP8266 Wireless Module with CP2102 UART/USB interface



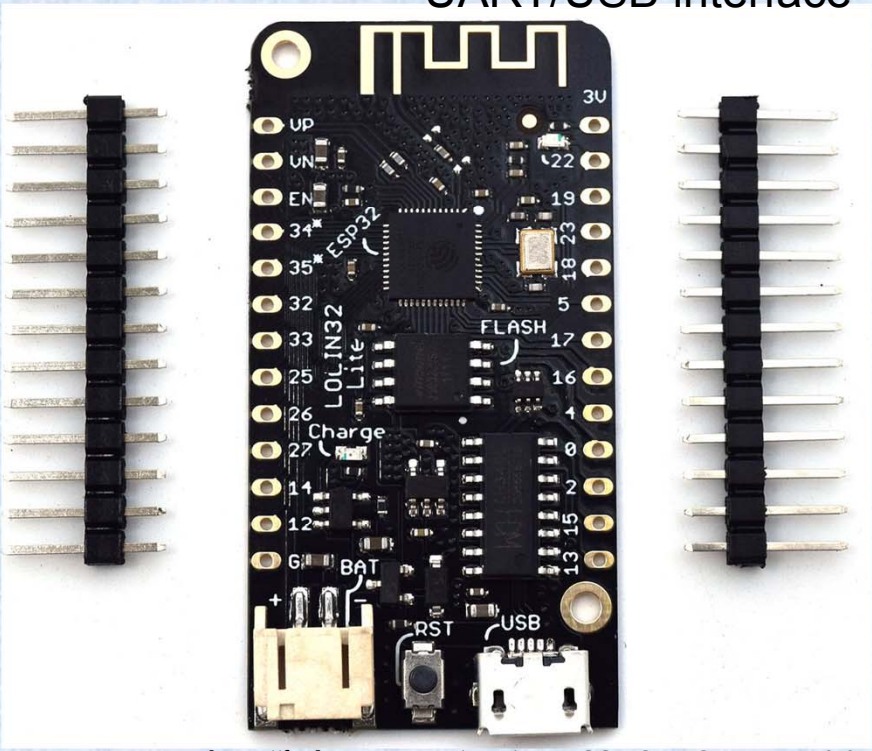
5.26 USD

http://www.banggood.com/NodeMcu-Lua-WIFI-Internet-Things-Development-Board-Based-ESP8266-CP2102-Wireless-Module-p-1097112.html

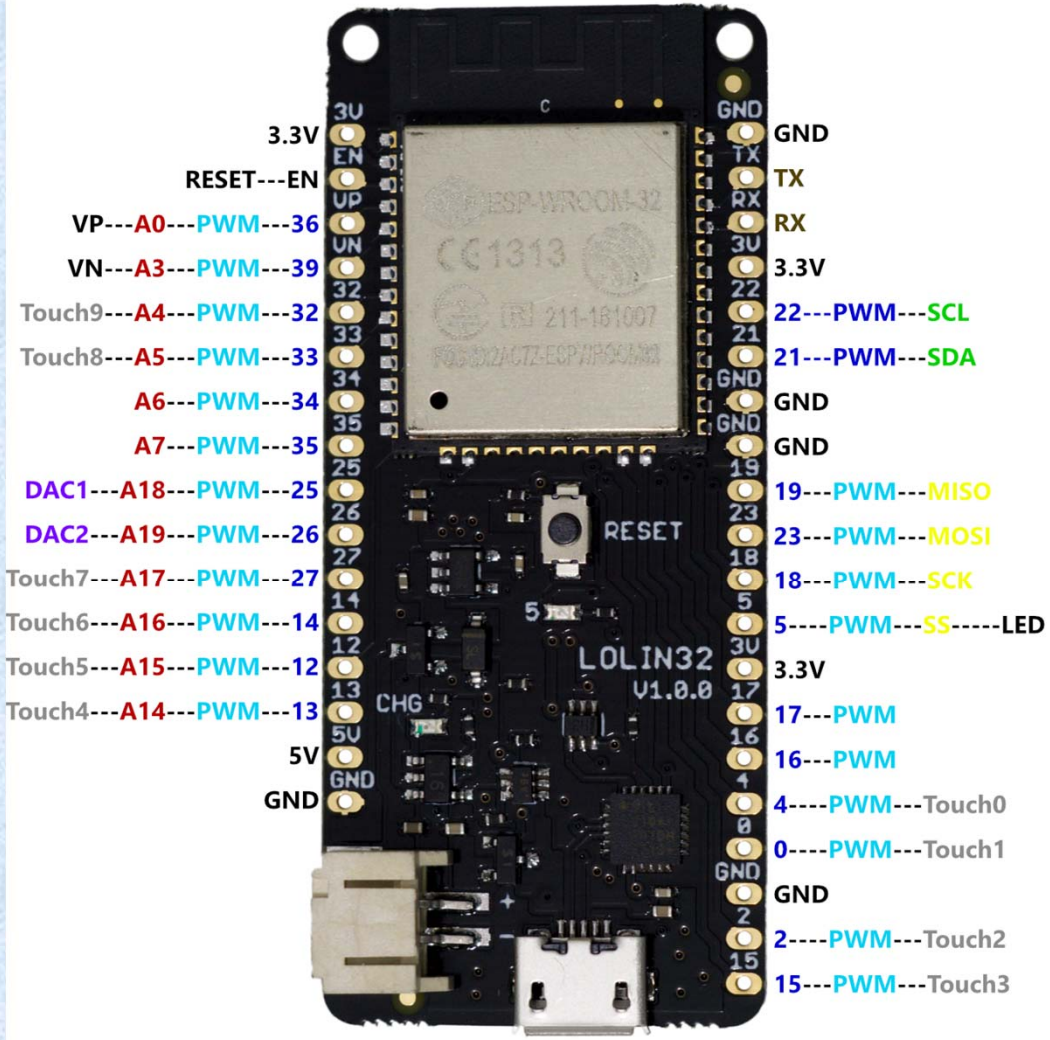
http://www.fais-le-toi-meme.fr/fr/electronique/tutoriel/programmes-arduino-executes-sur-esp8266-arduino-ide

Wemos Lolin32

- WIFI, BT BLE, 4 MB Flash, Lithium battery interface
- Internet of Things development board based on ESP32 with UART/USB interface



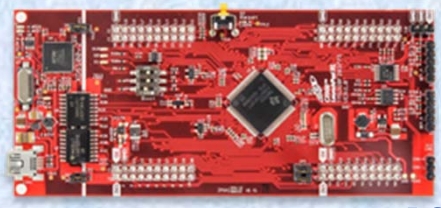
6 - 8 USD



https://fr.aliexpress.com/item/WEMOS-LOLIN32-Lite-V1-0-0-wifi-bluetooth-board-based-ESP-32-esp32-Rev1-MicroPython-4MB/32831394824.html
https://fr.aliexpress.com/item/4-MB-Flash-WEMOS-Lolin32-V1-0-0-WIFI-Bluetooth-Card-Based-ESP-32-ESP-WROOM/32823891656.html

LaunchPad C2000

http://www.ti.com/lsds/ti/tools-software/launchpads/launchpads.page#performance



Piccolo

LAUNCHXL-F28027F
Real time MCU with InstaSPIN-FOC motor control software.

- Featuring the 32-bit C2000 TMS320F28027F MCU @ **60MHz**, 64kB Flash / 12kB RAM. InstaSPIN-FOC enabled.
- Includes 8 PWM channels (4 high resolution), 12-bit 4.6 MSPS ADC, 3x 32-bit timers, temperature sensor, comparator.
- Motor control software in ROM implements InstaSPIN-FOC solutions for 3 phase motors.

17.00 USD

LAUNCHXL-F28069M
High performance controller with InstaSPIN motor control software.

- Featuring the 32-bit C2000 TMS320F28069M MCU @ **90MHz**, **FPU,** 256kB Flash / 96kB RAM. InstaSPIN-MOTION and InstaSPIN-FOC enabled.
- Includes 16 PWM channels (8 high resolution), 16 channel 12-bit 4.6 MSPS ADC, 3x 32-bit timers, 6 channel DMA, temperature sensor.
- Motor control software in ROM implements InstaSPIN-FOC and InstaSPIN-MOTION solutions for 3 phase motors.

24.99 USD

Delfino

LAUNCHXL-F28377S
High performance device for real-time control and signal processing applications.

- Featuring the 32-bit **Delfino** MCU TMS320F28377S: **@ 200 MHz**, **FPU,** 1MB Flash / 164kB RAM with ECC.
- Includes 24 PWM channels (16 high resolution), 4x 16/12-bit ADC, 3x 12-bit DAC, 3x eQEP modules, 169 GPIO pins.
- Includes CLA real-time control coprocessor to respond to peripheral triggers and service time-critical functions.

29.99 USD

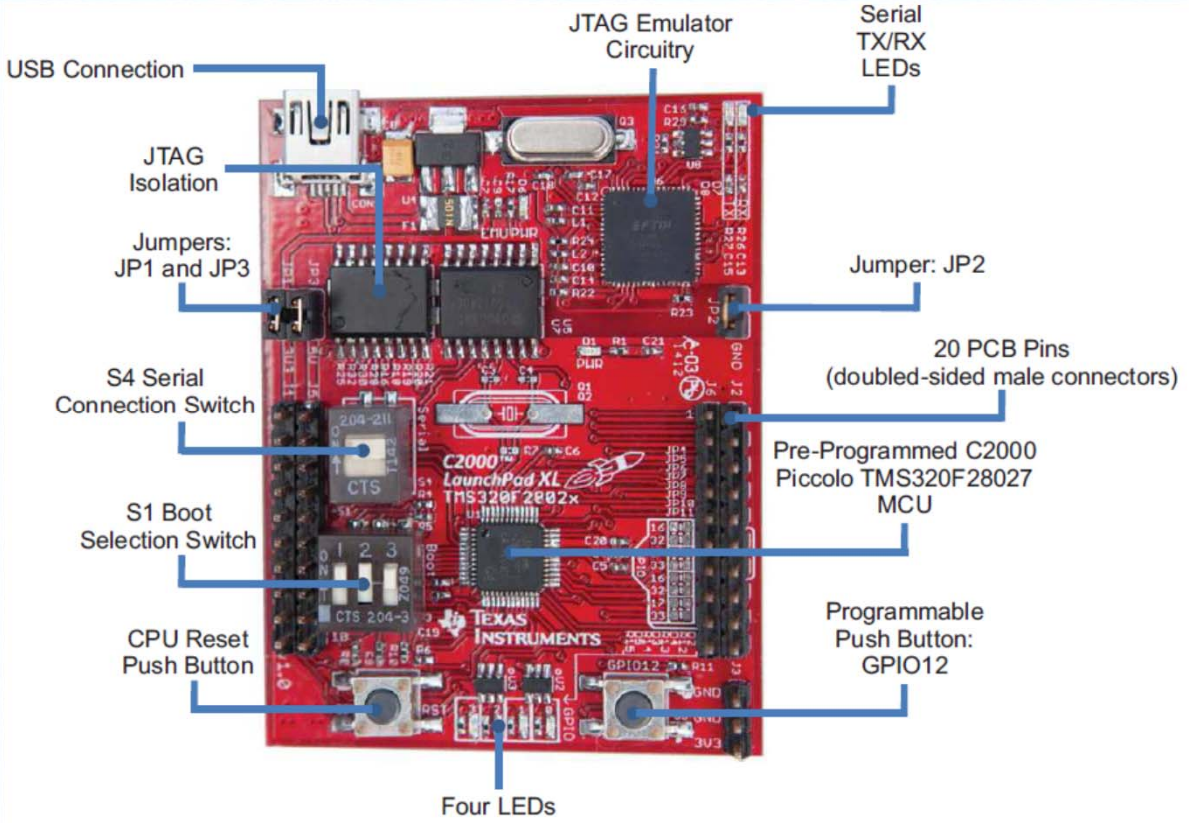
LAUNCHXL-F28379D
Dual Core MCU with highest level of peripheral integration in C2000 product family.

- Featuring the TMS320F28379D: **dual** 32-bit **Delfino** MCUs **@200MHz**, **FPU,** 1MB Flash/164KB RAM. Position Manager Enabled.
- 4 16/12-bit ADCs with on PCB instrumentational amplifiers. 3x 12-bit output DACs and 8 windowed comparators for asychronous output triggering.
- 24 PWMs (16 high resolution), 6 eCAP modules mapped to any GPIO,and 3 eQEP modules. 8 Sigma Delta Demodulator Inputs.

33.79 USD

LaunchPad C2000

- LaunchXL-F28027F QuickStartGuide sprz413.pdf
- LaunchXL-F28027F HW Exp guide spruhh2a.pdf
- http://www.ti.com/lsds/ti/tools-software/launchpads/launchpads.page#performance

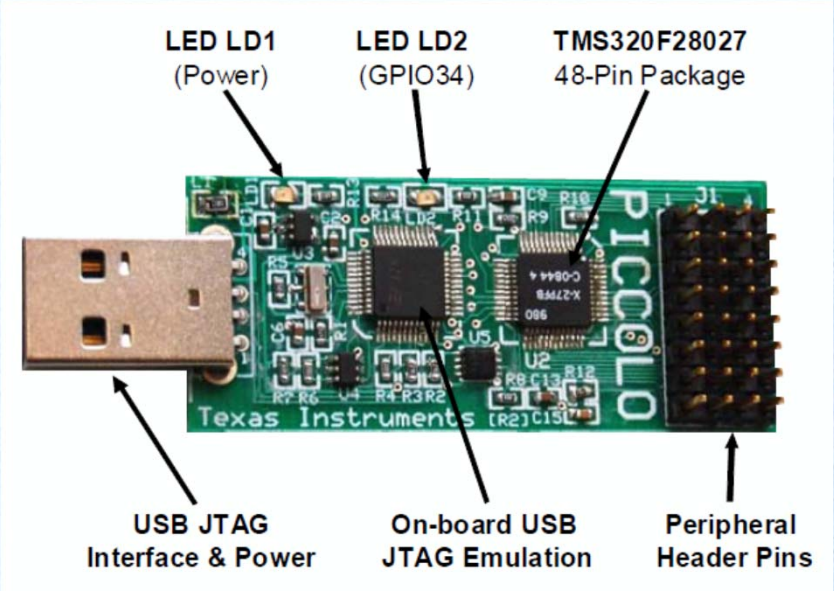


Real time MCU with InstaSPIN-FOC motor control software.

- Featuring the 32-bit C2000 TMS320F28027F MCU : @ 60MHz, 64kB Flash / 12kB RAM. InstaSPIN-FOC enabled. **No FPU**
- Includes 8 PWM channels (4 high resolution), 12-bit 4.6 MSPS ADC, 3x 32-bit timers, temperature sensor, comparator.
- Motor control software in ROM implements InstaSPIN-FOC solutions for 3 phase motors.

LaunchPad C2000

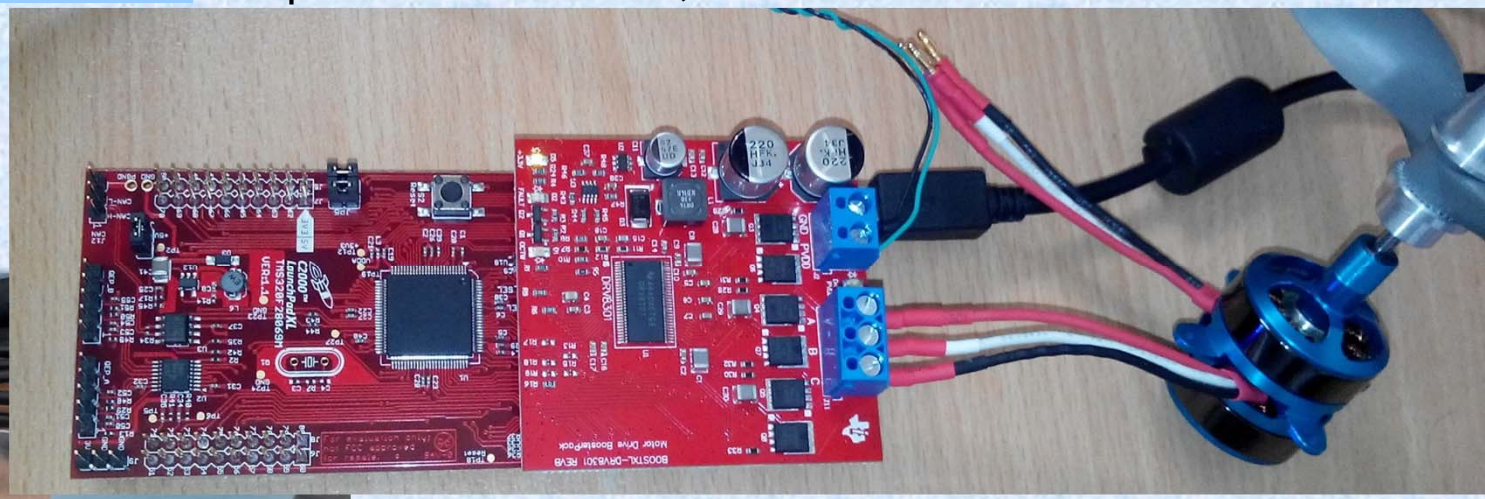
- Piccolo™ ControlSTICK sprt517b.pdf
- Explications

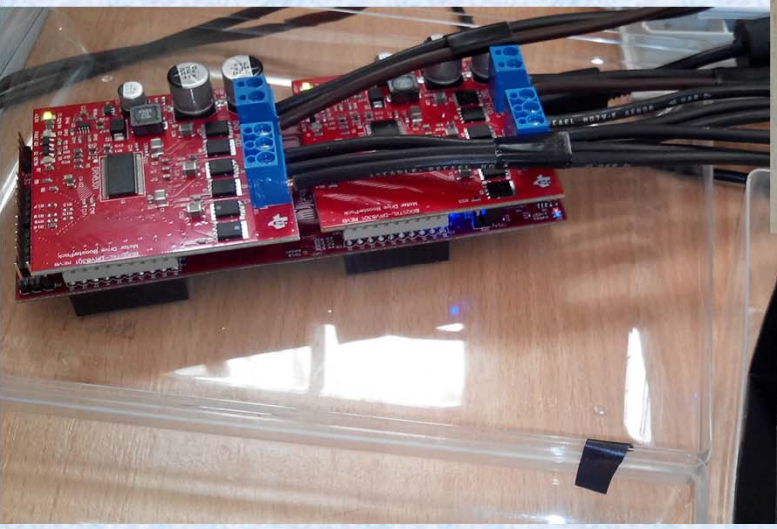


39.00 USD

LaunchPad C2000

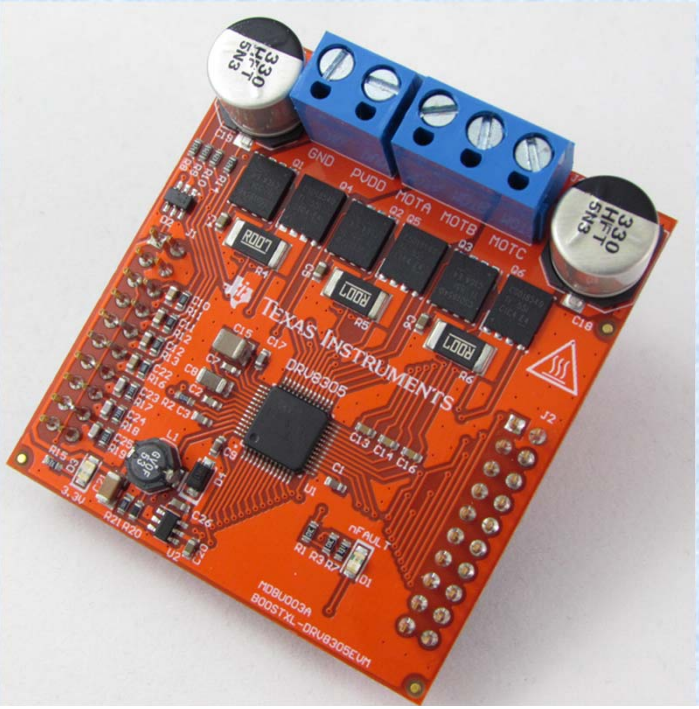
- LaunchXL-F28069M QuickStartGuide sprui02.pdf
- LaunchXL-F28069M User s guide sprui11.pdf
- BoosterPack plug-in modules
  - BOOSTXL-DRV8301 : tri-phase inverter 10A, 6-24V

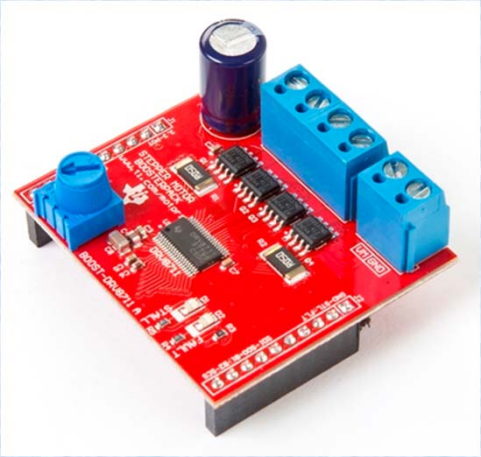




LaunchPad C2000

- BoosterPack plug-in modules (XL : 2 rangés x 2)
  - BOOSTXL-DRV8301 : Onduleur triphasé 10A RMS, 6 - 24V
  - BOOSTXL-8305EVM : Onduleur triphasé 15A RMS, 4.4 - 45V
  - BOOST-DRV8711 : Bipolar stepper motor drive 4.5A, 8-52V
  - Nokia 5110 LCD BoosterPack/Breakout PCB

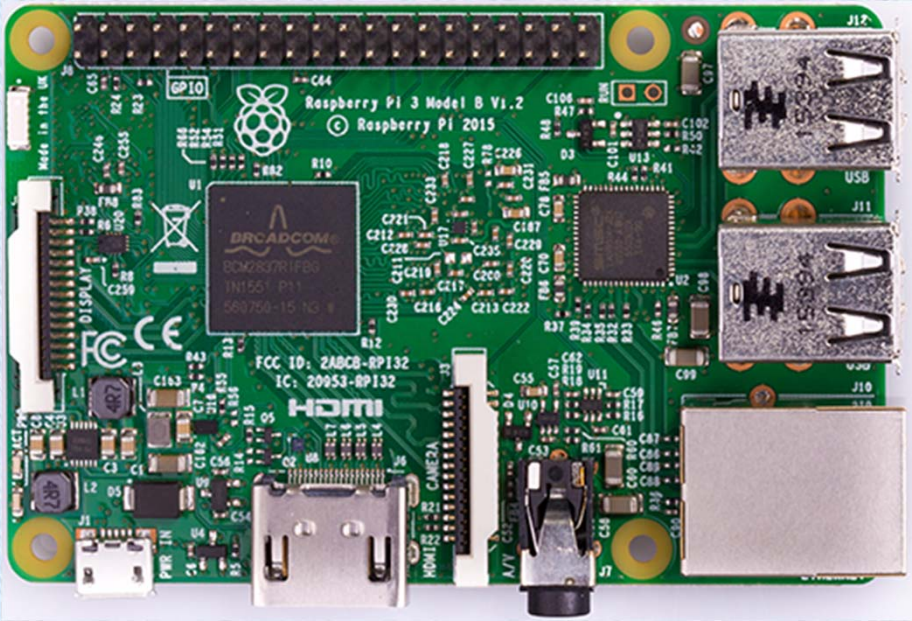




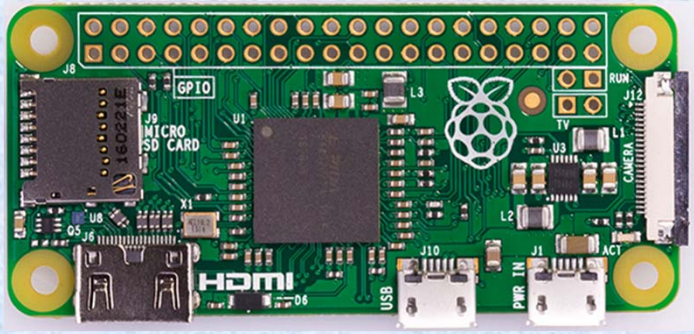
49.00 USD

Rasbperry Pi

- Pi 3 : 1.2GHz 64-bit quad-core ARMv8 CPU; Wifi 802.11n, Bluetooth 4.1 BLE, 1GB RAM, 4 USB ports, 40 GPIO pins, HDMI, Ethernet, 3.5mm audio / c-video, Camera interface (CSI), Display interface (DSI), Micro SD card slot
- Pi zero : 1Ghz, Single-core CPU, 512MB RAM, mini HDMI and USB OTG, Micro USB power, HAT-compatible 40-pin header



48 €



5 USD

https://www.raspberrypi.org/magpi-issues/ https://www.raspberrypi.org/products/pi-zero/

Developement tools

- Arduino IDE 1.81
- MPLAB X 3.50 + Monitor : Putty
- Code Composer Studio (CCS) 7
- GCC (Raspberry Pi)
- Logic analyzer (SPI, UART, I2C, CAN)

https://www.arduino.cc/en/Main/Software
http://www.microchip.com/mplab/mplab-x-ide
http://www.ti.com/tool/CCSTUDIO

Tools

- Arduino IDE
- Monitor

COM11

Send

```
closing connection
AcX = 4296 | AcY = 248 | AcZ = -16916 | Tmp = 27.73 | GyX = -378 | GyY = 79 | GyZ = 46
AcX = 4484 | AcY = 532 | AcZ = -16500 | Tmp = 27.87 | GyX = -612 | GyY = 339 | GyZ = 84
AcX = 4852 | AcY = 712 | AcZ = -16624 | Tmp = 28.06 | GyX = -345 | GyY = 53 | GyZ = 72
AcX = 4824 | AcY = 736 | AcZ = -16788 | Tmp = 28.11 | GyX = -321 | GyY = 31 | GyZ = 43
AcX = 4888 | AcY = 692 | AcZ = -16748 | Tmp = 28.29 | GyX = -383 | GyY = 117 | GyZ = 65
connecting to [redacted]
Requesting URL: [redacted]?t=1&d=15&x=4888&y=692&z=-16748&tm=28.29&h=-383
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 20 Jan 2017 23:01:40 GMT
Content-Type: text/html
Content-Length: 0
Connection: close
X-Powered-By: PHP/5.4.44


closing connection
AcX = 4828 | AcY = 728 | AcZ = -16748 | Tmp = 28.58 | GyX = -373 | GyY = 64 | GyZ = 65
AcX = 4744 | AcY = 700 | AcZ = -16700 | Tmp = 28.58 | GyX = -364 | GyY = 96 | GyZ = 40
AcX = 4900 | AcY = 720 | AcZ = -16816 | Tmp = 28.77 | GyX = -326 | GyY = 68 | GyZ = 37
AcX = 4776 | AcY = 724 | AcZ = -16740 | Tmp = 28.81 | GyX = -346 | GyY = 35 | GyZ = 40
AcX = 4660 | AcY = 704 | AcZ = -16764 | Tmp = 29.09 | GyX = -343 | GyY = 53 | GyZ = 56
connecting [redacted]
Requesting URL: [redacted]php?t=1&d=20&x=4660&y=704&z=-16764&tm=29.09&h=-343
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 20 Jan 2017 23:01:46 GMT
Content-Type: text/html
Content-Length: 0
Connection: close
X-Powered-By: PHP/5.4.44


closing connection
AcX = 4892 | AcY = 612 | AcZ = -16544 | Tmp = 29.28 | GyX = -383 | GyY = 92 | GyZ = 35
AcX = 4884 | AcY = 740 | AcZ = -16720 | Tmp = 29.38 | GyX = -383 | GyY = 122 | GyZ = 55
```

Autoscroll | Both NL & CR | 115200 baud

https://www.arduino.cc/en/Main/Software

IMUWifi | Arduino 1.8.1

File Edit Sketch Tools Help

IMUWifi

```
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
#endif

  // pendant ce temps, on config IMU
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B);  // PWR_MGMT_1 register
  Wire.write(0);     // set to zero (wakes up the MPU-6050)
  Wire.endTransmission(true);
}

int Count = 0;
int WifiDS = 0;

void loop() {
  delay(1000);  // Check IMU ttes les 1 seconde
  WifiDS++;
  Count++;

  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B);  // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr,14,true);  // request a total of 14 registers
  AcX=Wire.read()<<8|Wire.read();  // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AcY=Wire.read()<<8|Wire.read();  // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AcZ=Wire.read()<<8|Wire.read();  // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
  Tmp=Wire.read()<<8|Wire.read();  // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
  TmpC = Tmp/340.00+36.53;
  GyX=Wire.read()<<8|Wire.read();  // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
  GyY=Wire.read()<<8|Wire.read();  // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
  GyZ=Wire.read()<<8|Wire.read();  // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
  Serial.print("AcX = "); Serial.print(AcX);
  Serial.print(" | AcY = "); Serial.print(AcY);
  Serial.print(" | AcZ = "); Serial.print(AcZ);
  Serial.print(" | Tmp = "); Serial.print(TmpC);  //equation for temperature in degrees C from datas
  Serial.print(" | GyX = "); Serial.print(GyX);
  Serial.print(" | GyY = "); Serial.print(GyY);
  Serial.print(" | GyZ = "); Serial.println(GyZ);
```

Done compiling.

```
C:\Users\baghli\AppData\Local\Arduino15\packages\esp8266\hardware\esp8266\2.3.0\libraries\Wire/Wire.
    size_t requestFrom(uint8_t address, size_t size, bool sendStop);
           ^

Sketch uses 237249 bytes (22%) of program storage space. Maximum is 1044464 bytes.
Global variables use 33156 bytes (40%) of dynamic memory, leaving 48764 bytes for local variables. M
```

18 | WeMos D1 R2 & mini, 80 MHz, 921600, 4M (3M SPIFFS) on COM11

Tools MPLAB X

MPLAB X IDE v3.50 -

File Edit View Navigate Source Refactor Run Debug Team Tools Window Help

Projects

- Header Files
- Important Files
- Linker Files
- Source Files
- Libraries
- Loadables
- essai33EP

main.c | interrupts.c | system.c | myuart.c | main.c

Source History

```
#include "main.h"
#include "common.h"
#include "myuart.h" // fichier perso
#include "sht11.h"
#include "SST26VF032B.h"
#include "pleds.h"
#include "sst_helper.h"
//#include "HWP EX16_MRF24W XC16.h"
#include "SNMPv3.h"


#pragma config AWRP = OFF        // Auxiliary Segment Write-protect bit (Auxiliary program memory is not write-protected)
#pragma config APL = OFF         // Auxiliary Segment Code-protect bit (Aux Flash Code protect is disabled)
#pragma config APLK = OFF        // Auxiliary Segment Key bits (Aux Flash Write Protection and Code Protection is Disabled)
// FICD
#pragma config ICS = PGD1        // ICD Communication Channel Select bits (Communicate on PGEC1 and PGED1)
#pragma config RSTPRI = PF       // Reset Target Vector Select bit (Device will obtain reset instruction from Primary flash)
#pragma config JTAGEN = OFF      // JTAG Enable bit (JTAG is disabled)
// FPOR
#pragma config FPWRT = PWR128    // Power-on Reset Timer Value Select bits (128ms)
#pragma config BOREN = ON        // Brown-out Reset (BOR) Detection Enable bit (BOR is enabled)
#pragma config ALTI2C1 = OFF     // Alternate I2C pins for I2C1 (SDA1/SCK1 pins are selected as the I/O pins for I2C1)
// FWDT
#pragma config WDTPOST = PS32768     // Watchdog Timer Postscaler Bits (1:32,768)
#pragma config WDTPRE = PR128        // Watchdog Timer Prescaler bit (1:128)
#pragma config PLLKEN = ON           // PLL Lock Wait Enable bit (Clock switch to PLL source will wait until the PLL lock signal is valid.)
#pragma config WINDIS = OFF          // Watchdog Timer Window Enable bit (Watchdog Timer in Non-Window mode)
#pragma config FWDTEN = OFF          // Watchdog Timer Enable bit (Watchdog timer enabled/disabled by user software)
// FOSC -> PLL 10MHz
#pragma config POSCMD = XT       // Primary Oscillator Mode Select bits (XT Crystal Oscillator Mode)
#pragma config OSCIOFNC = OFF    // OSC2 Pin Function bit (OSC2 is clock output)      // non utilisé en XT mode
#pragma config IOL1WAY = OFF     // Peripheral pin select configuration (Allow multiple reconfigurations)
#pragma config FCKSM = CSDCMD    // Clock Switching Mode bits (Both Clock switching and Fail-safe Clock Monitor are disabed
// FOSCSEL
#pragma config FNOSC = PRIPLL    // Initial Oscillator Source Selection Bits (Primary Oscillator (XT, HS, EC) with PLL)
#pragma config IESO = OFF        // Two-speed Oscillator Start-up Enable bit (Start up with user-selected oscillator source)
// FGS
#pragma config GWRP = OFF        // General Segment Write-Protect bit (General Segment may be written)
#pragma config GSS = OFF         // General Segment Code-Protect bit (General Segment Code protect is disabled)
#pragma config GSSK = OFF        // General Segment Key bits (General Segment Write Protection and Code Protection is Disabled)
/*
// DSPIC33EP512MU810 Configuration Bit Settings
```

Output | Variables

Project Loading Warning | Configuration Loading Error | PICkit 3 | (Build, Load, ...) | Debugger Console

```
Running

Halting...

Target Halted
```

Tools Code Composer Studio

ti - Debug - Example_F2802xLaunchPadDemo/Example_F2802xLaunchPadDemo.c - Code Composer Studio

File Edit Source Refactor View Navigate Search Project Run Window Help

Example_F2802xLaunchPadDemo.c

```c
276
277 void main()
278 {
279     volatile int status = 0;
280     volatile FILE *fid;
281
282     CPU_Handle myCpu;
283     PLL_Handle myPll;
284     WDOG_Handle myWDog;
285
286     // Initialize all the handles needed for this application
287     myAdc = ADC_init((void *)ADC_BASE_ADDR, sizeof(ADC_Obj));
288     myClk = CLK_init((void *)CLK_BASE_ADDR, sizeof(CLK_Obj));
289     myCpu = CPU_init((void *)NULL, sizeof(CPU_Obj));
290     myFlash = FLASH_init((void *)FLASH_BASE_ADDR, sizeof(FLASH_Obj));
291     myGpio = GPIO_init((void *)GPIO_BASE_ADDR, sizeof(GPIO_Obj));
292     myPie = PIE_init((void *)PIE_BASE_ADDR, sizeof(PIE_Obj));
293     myPll = PLL_init((void *)PLL_BASE_ADDR, sizeof(PLL_Obj));
294     mySci = SCI_init((void *)SCIA_BASE_ADDR, sizeof(SCI_Obj));
295     myWDog = WDOG_init((void *)WDOG_BASE_ADDR, sizeof(WDOG_Obj));
296
297     // Perform basic system initialization
298     WDOG_disable(myWDog);
299     CLK_enableAdcClock(myClk);
300     (*Device_cal)();
301
302     //Select the internal oscillator 1 as the clock source
303     CLK_setOscSrc(myClk, CLK_OscSrc_Internal);
304
305     // Setup the PLL for x10 /2 which will yield 50Mhz = 10Mhz * 10 / 2
306     PLL_setup(myPll, PLL_Multiplier_12, PLL_DivideSelect_ClkIn_by_2);
307
308     // Disable the PIE and all interrupts
309     PIE_disable(myPie);
310     PIE_disableAllInts(myPie);
311     CPU_disableGlobalInts(myCpu);
312     CPU_clearIntFlags(myCpu);
313
314     // If running from flash copy RAM only functions to RAM
315 #ifdef _FLASH
316     memcpy(&RamfuncsRunStart, &RamfuncsLoadStart, (size_t)&RamfuncsLoadSize);
317 #endif
318
319     // Initalize GPIO
320     // Enable XCLOCKOUT to allow monitoring of oscillator 1
```

Terminal — Serial COM16 (21/01/17 00:41)

```
 ==========================                    II      ========================
 | Texas Instruments      |                            | Press S3 to begin      |
Current Temperature:nchpad |                            | demo                   |
37 Celcius = Ref + 0 37 Celcius = Ref + 0 37 Celcius = Ref + 0 37 Celcius = Ref + 0 37
 Ref + 0 37 Celcius = Ref + 0 38 Celcius = Ref + 1 37 Celcius = Ref + 0 38 Celcius = R
Celcius = Ref + 0 37 Celcius = Ref + 0 38 Celcius = Ref + 1 39 Celcius = Ref + 2 38 Ce
f + 1 38 Celcius = Ref + 1 37 Celcius = Ref + 0 38 Celcius = Ref + 1 38 Celcius = Ref
cius = Ref + 1 38 Celcius = Ref + 1 39 Celcius = Ref + 2 39 Celcius = Ref + 2 39 Celci
 2 38 Celcius = Ref + 1 39 Celcius = Ref + 2 38 Celcius = Ref + 1 38 Celcius = Ref + 1
s = Ref + 1 38 Celcius = Ref + 1 38 Celcius = Ref + 1 39 Celcius = Ref + 2 39 Celcius
38 Celcius = Ref + 1 39 Celcius = Ref + 2 38 Celcius = Ref + 1
```

Expression	Type	Valu
encHandle->enc_elec_angle	unknown	iden
gMotorVars.Flag_enableSys	unknown	iden
gMotorVars.SpeedRef_krpm	unknown	iden
gMotorVars.SpeedQEP_krpm	unknown	iden
gMotorVars.Speed_krpm	unknown	iden
gMotorVars.Flag_Run_Identify	unknown	iden
EnableFlag	unknown	iden
SpeedRef	unknown	iden
speed1.SpeedRpm	unknown	iden
RunMotor	unknown	iden
currentTemp	int	38
referenceTemp	int	37
encHandle->speed_lpf_out	unknown	iden
enc	unknown	iden
temp	unknown	iden
delta_enc	unknown	iden
enc_val	unknown	iden
t_AngleQEP	unknown	iden
t_SpeedQEP_krpm	unknown	iden
t_Speed_krpm	unknown	iden
thetas_angle_pu	unknown	iden
K_V_f	unknown	iden
WmMesure_int32	unknown	iden
WmMesure	unknown	iden
SpeedQEP_pu	unknown	iden
OldOut	unknown	iden
CountIndex	unknown	iden
CountIndexMax	unknown	iden
limit_integral_range	unknown	iden
ADC_IntNumber_1	unknown	iden
ADC_IntNumber_1	unknown	iden
currentTemp	int	38
referenceTemp	int	37
j	unknown	iden
j	unknown	iden

Console Tasks Problems Executables Debug

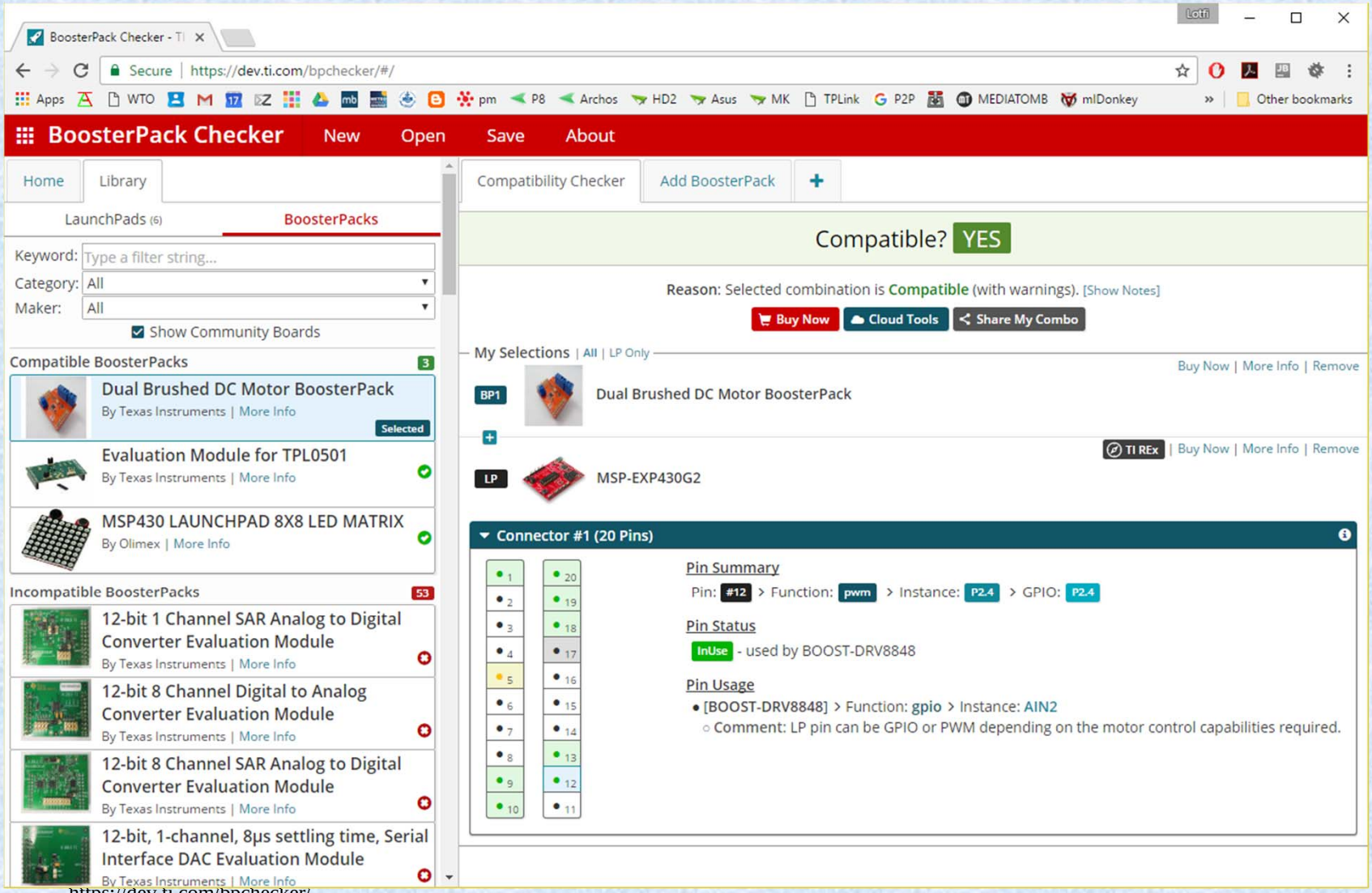
Example_F2802xLaunchPadDemo [Code Composer Studio - Device Debugging]

Texas Instruments XDS100v2 USB Debug Probe_0/C28xx (Suspended)

$C:/controlsuite_gitorious_dev/device_support/f2802x/version/f2802x_common/source/F2802x_asmfuncs.asm:110:130$() at F2802x_asmfuncs.asm:110 0x00801C ($C:/controlsuite_gitorious_dev/device_support/f2802x/version/f2802x_common/source/F2802x_asmfuncs.asm:110:13

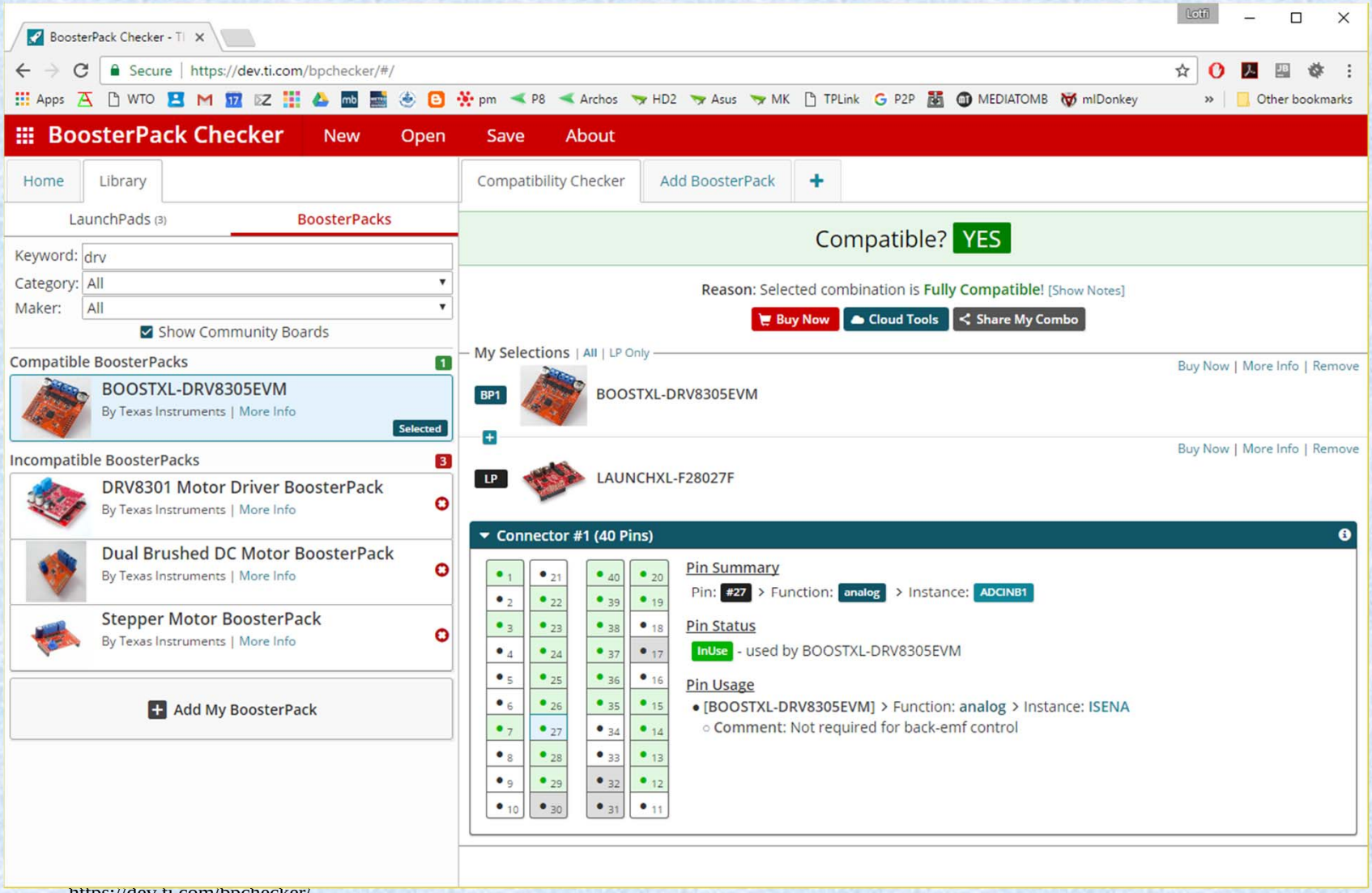
Connected - Encoding: Default (ISO-8859-1)

Tools BoosterPack Checker



https://dev.ti.com/bpchecker/

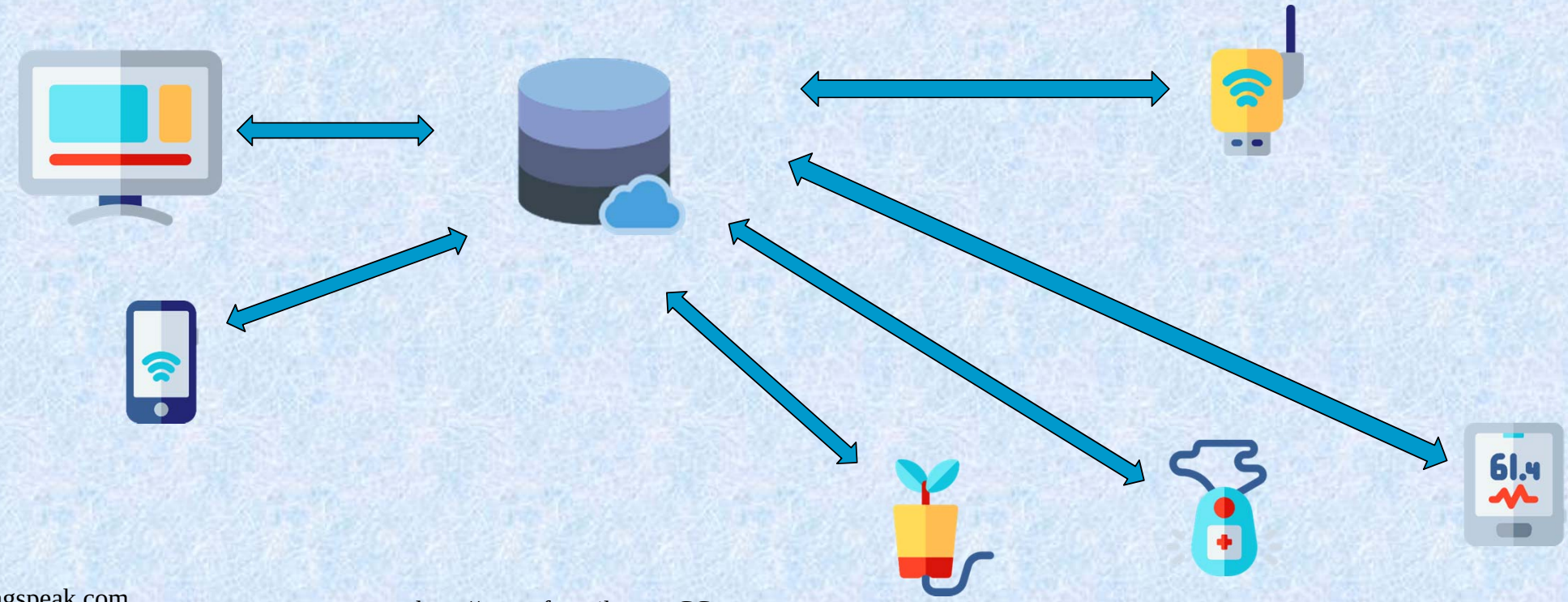
Tools BoosterPack Checker F28027F



https://dev.ti.com/bpchecker/

Internet of Things

- Microcontroller + Internet connectivity (Wifi, RJ45, 3G)
- Cloud database (MySQL, Firebase, ThingSpeak)

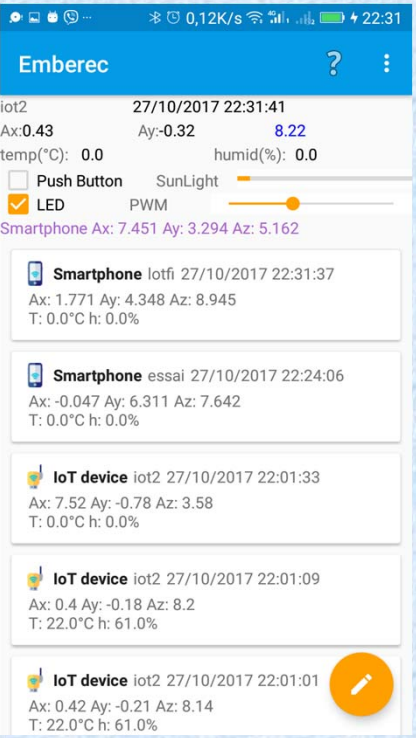


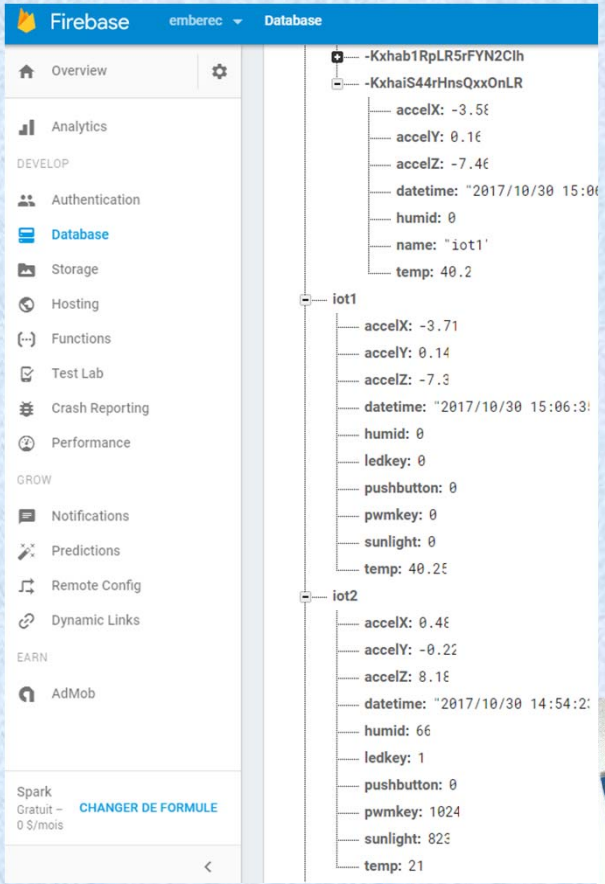
http://www.freepik.com CC

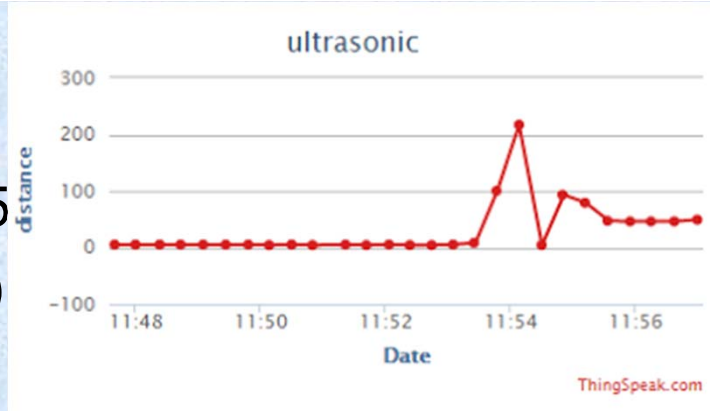
https://thingspeak.com
https://firebase.google.com
https://baghli.blogspot.com/2015/12/esp8266-wifi-ultrasonic-distance-logger.html

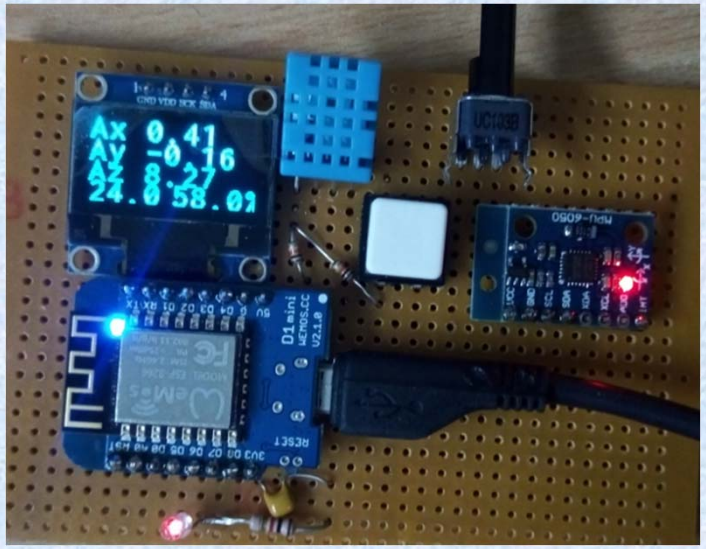
Internet of Things

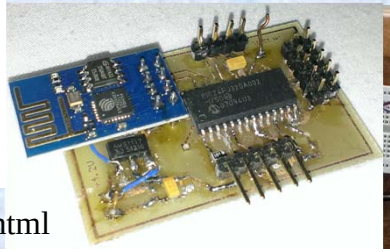
- Microcontroller + Internet connectivity (Wifi, RJ45
- Cloud database (MySQL, Firebase, ThingSpeak)

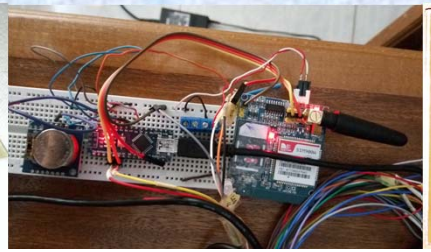


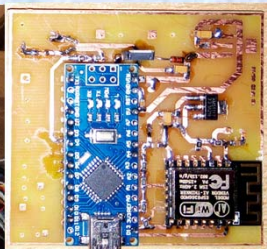








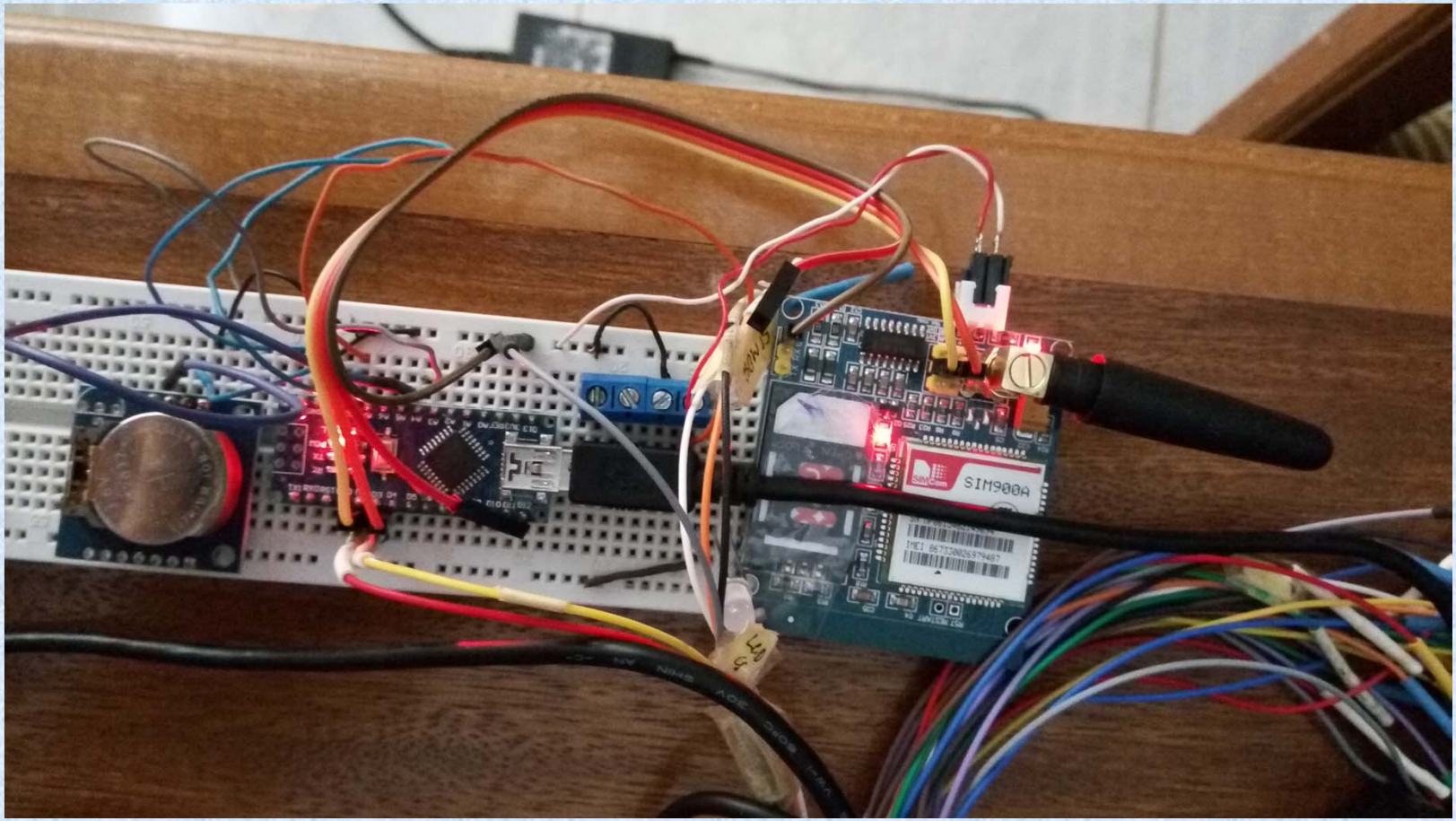




https://thingspeak.com
https://firebase.google.com
https://baghli.blogspot.com/2015/12/esp8266-wifi-ultrasonic-distance-logger.html

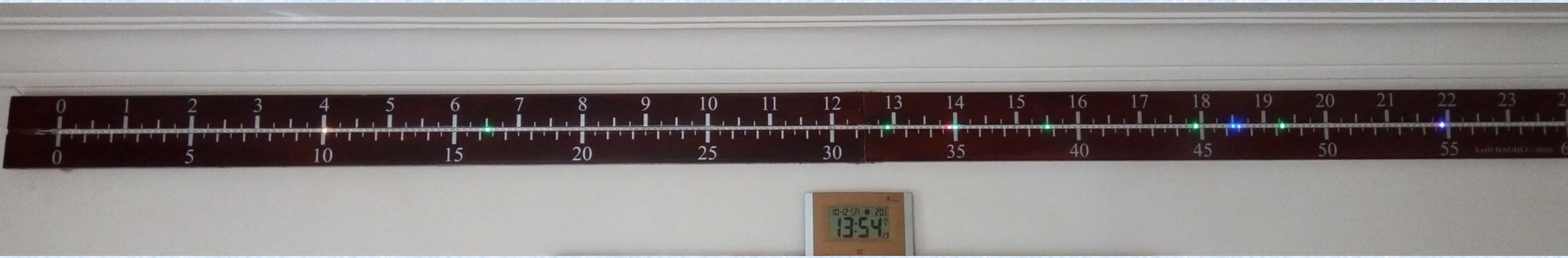
Examples

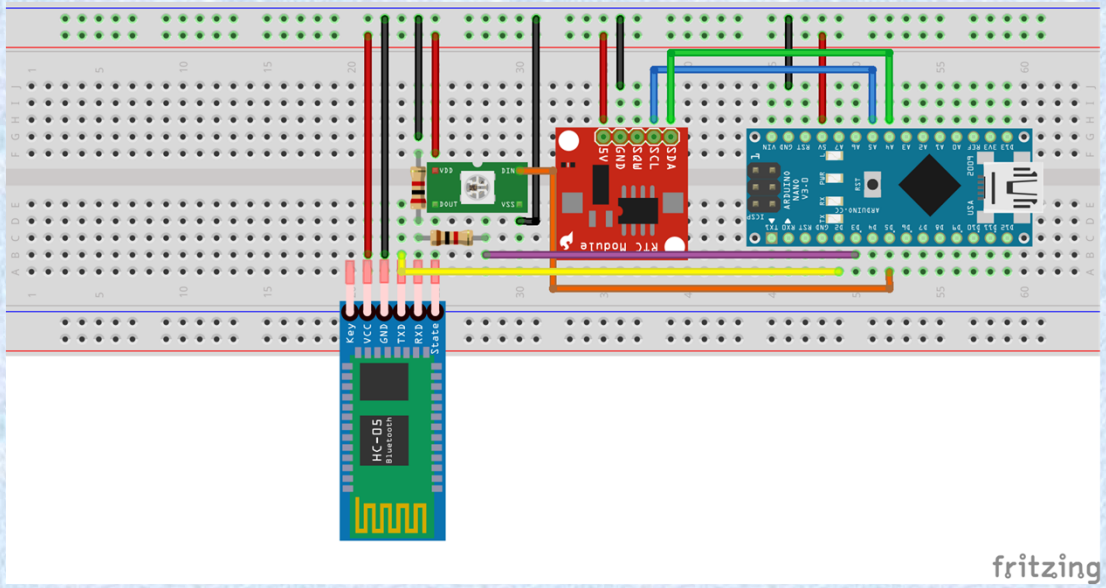
Arduino nano V3, RTC + Li rechargable, SIM900A



Examples

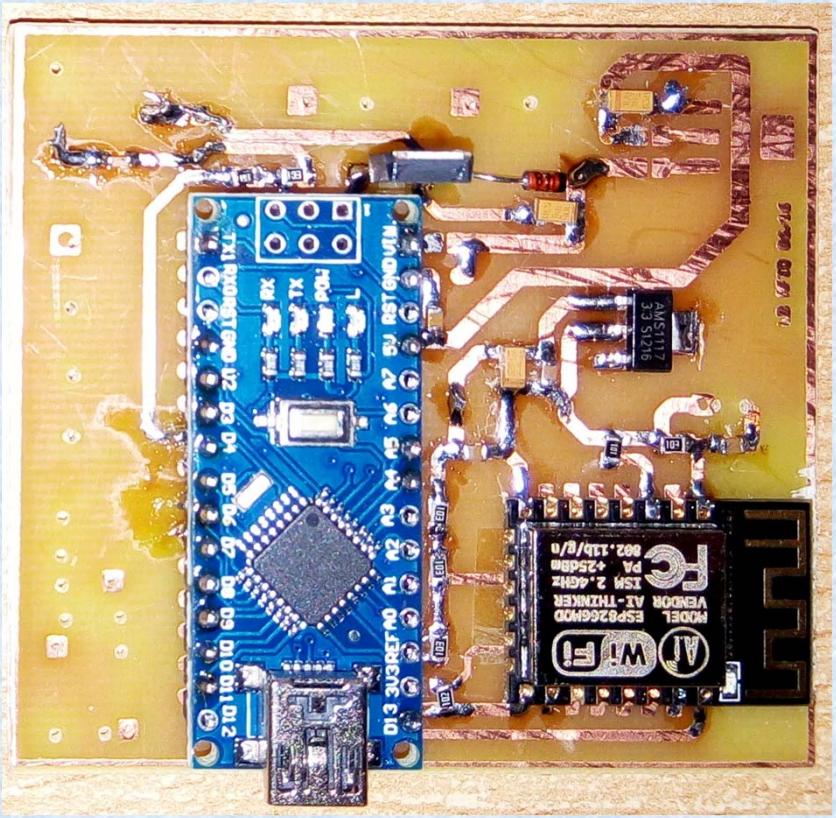
Arduino nano V3, RTC + Li rechargable, BT HC05, LED RGB WS2812





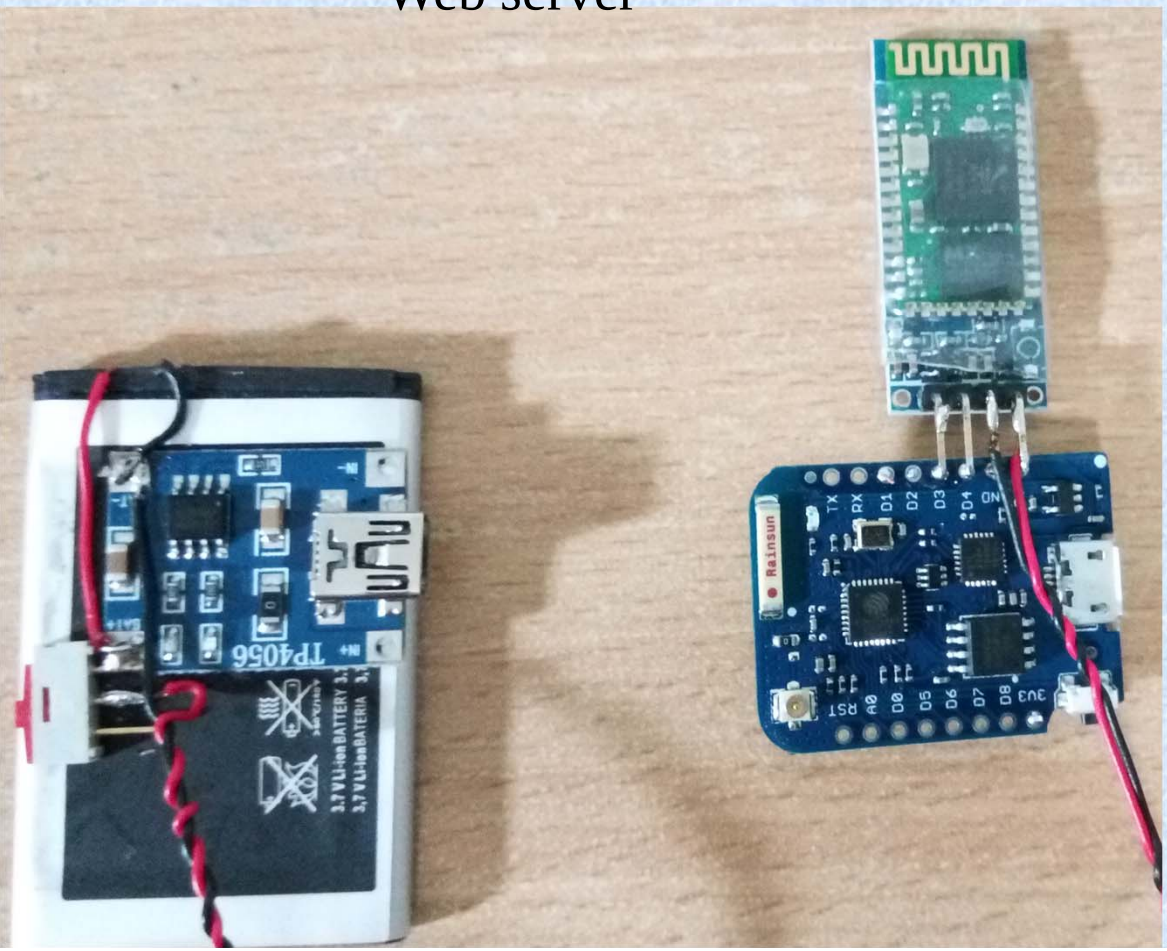
Examples

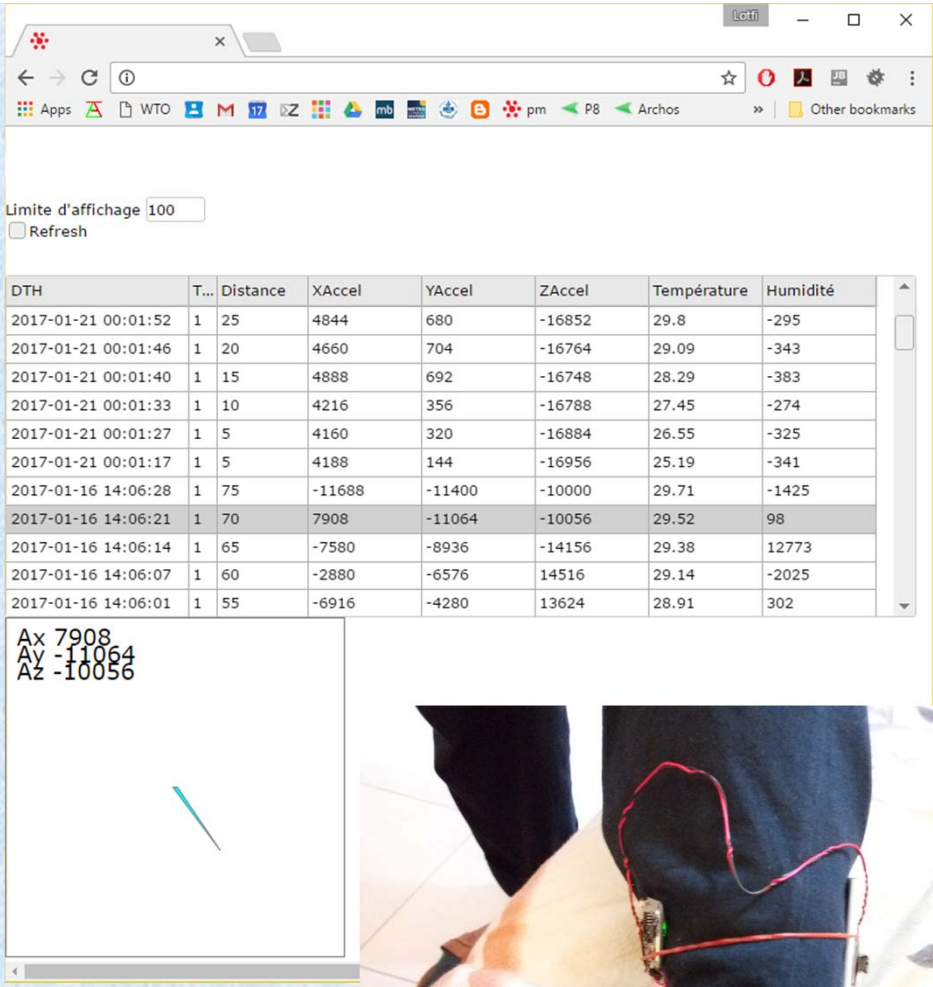
Arduino nano V3, Wifi ESP8266 pcb antenna, MOSFET, DRL, LDO 3.3V, RFID



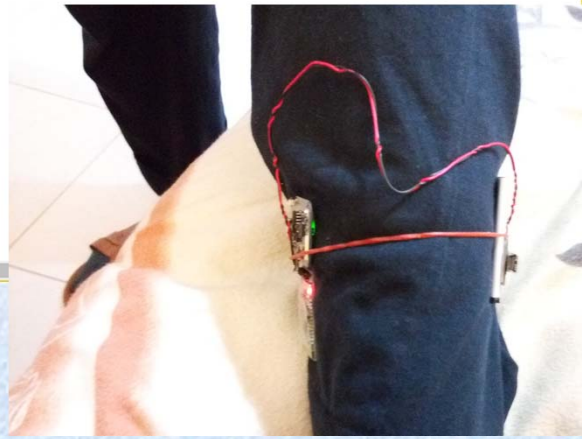
Examples

Wemos D1 mini pro (Wifi ESP8266 pcb antenna), BT, Li-Ion+chargeur
Web server



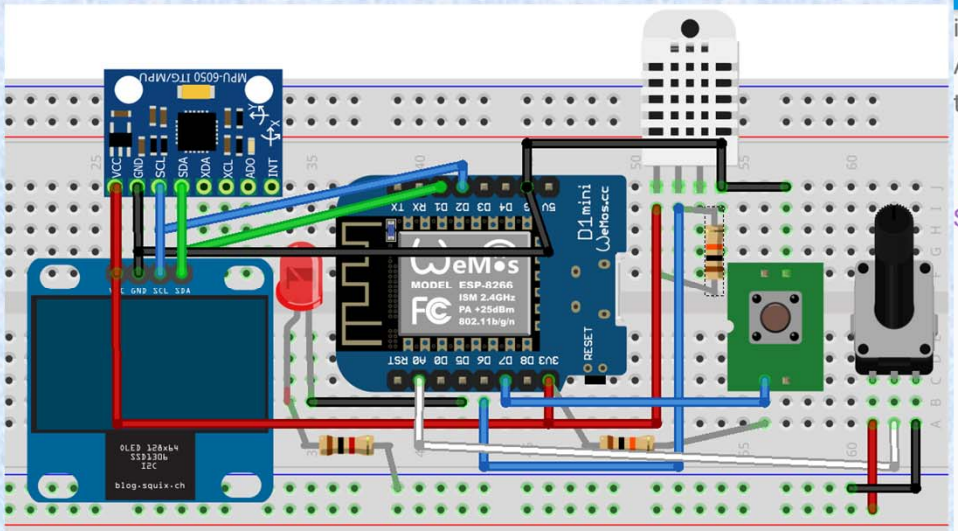


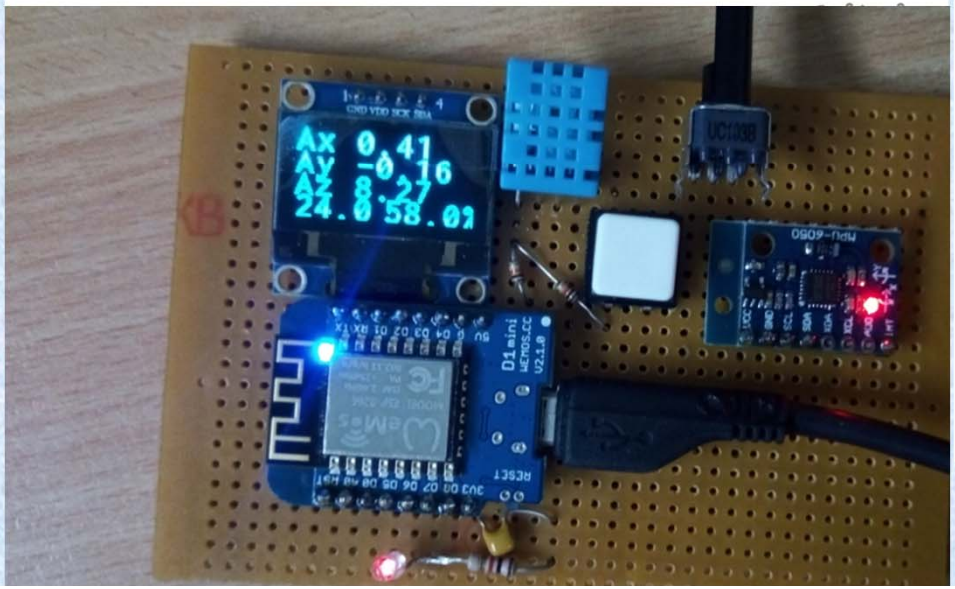
DTH	T...	Distance	XAccel	YAccel	ZAccel	Température	Humidité
2017-01-21 00:01:52	1	25	4844	680	-16852	29.8	-295
2017-01-21 00:01:46	1	20	4660	704	-16764	29.09	-343
2017-01-21 00:01:40	1	15	4888	692	-16748	28.29	-383
2017-01-21 00:01:33	1	10	4216	356	-16788	27.45	-274
2017-01-21 00:01:27	1	5	4160	320	-16884	26.55	-325
2017-01-21 00:01:17	1	5	4188	144	-16956	25.19	-341
2017-01-16 14:06:28	1	75	-11688	-11400	-10000	29.71	-1425
2017-01-16 14:06:21	1	70	7908	-11064	-10056	29.52	98
2017-01-16 14:06:14	1	65	-7580	-8936	-14156	29.38	12773
2017-01-16 14:06:07	1	60	-2880	-6576	14516	29.14	-2025
2017-01-16 14:06:01	1	55	-6916	-4280	13624	28.91	302

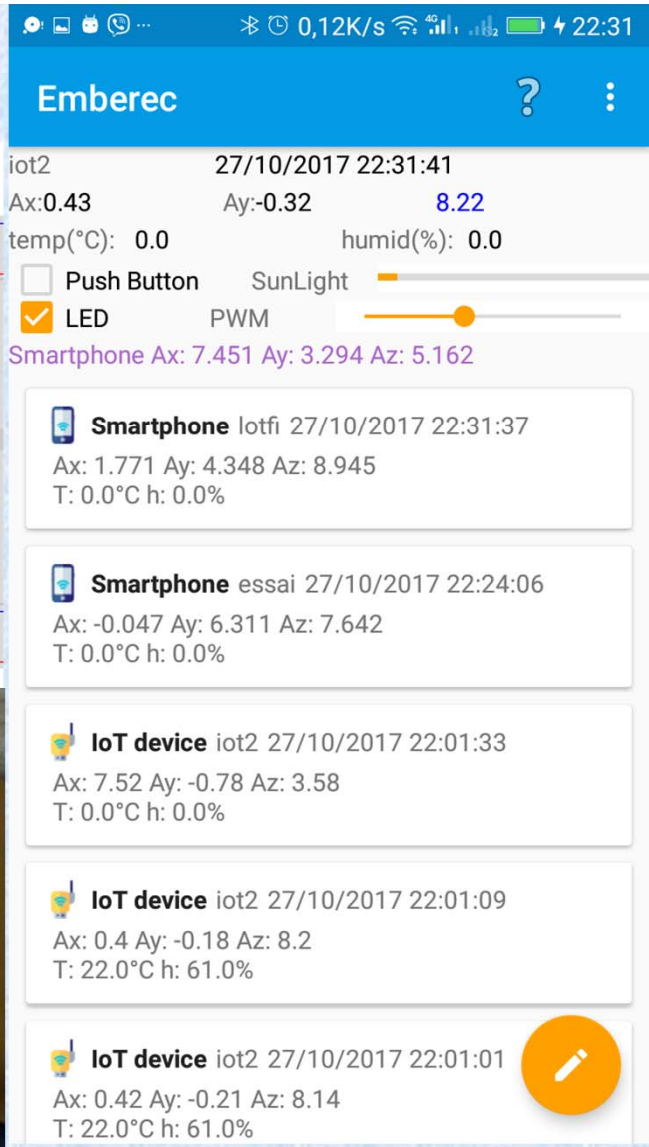


Examples

- WEMOS 8266
- OLED 0.98” I2C
- IMU MPU6050
- DHT 11
- LED, PB, Pot.
- Firebase Google database
- Android App: Emberec
- https://play.google.com/store/apps/details?id=com.embesystems.emberec

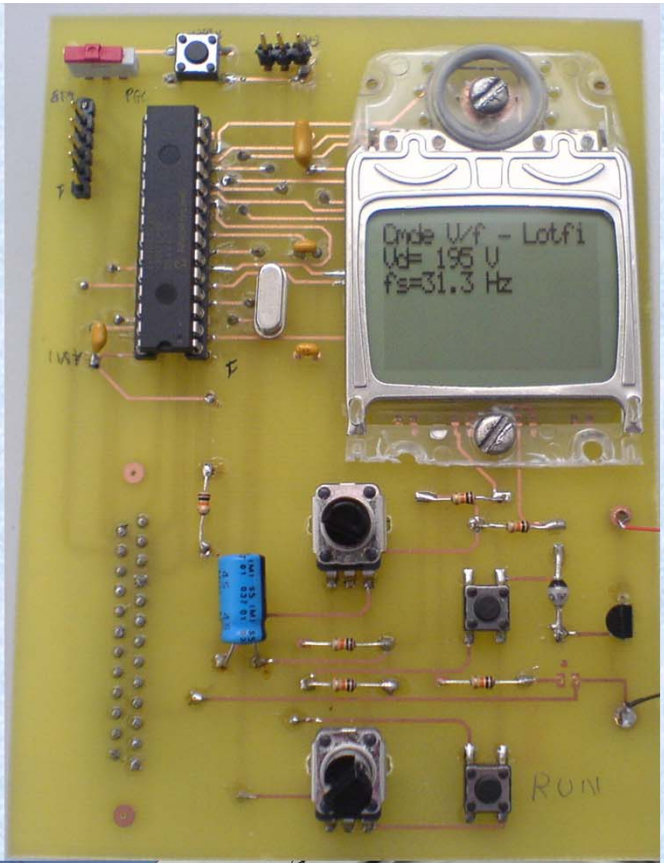


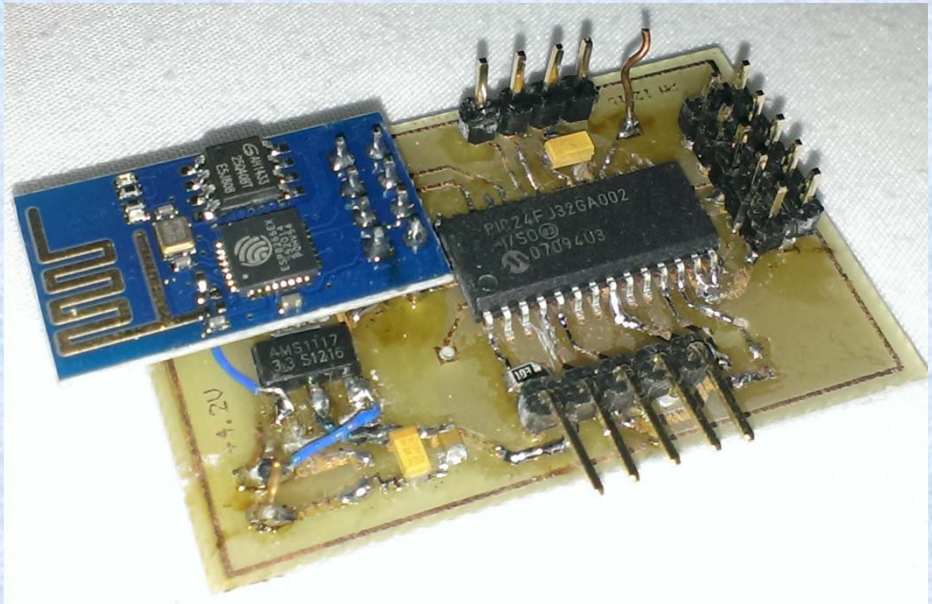


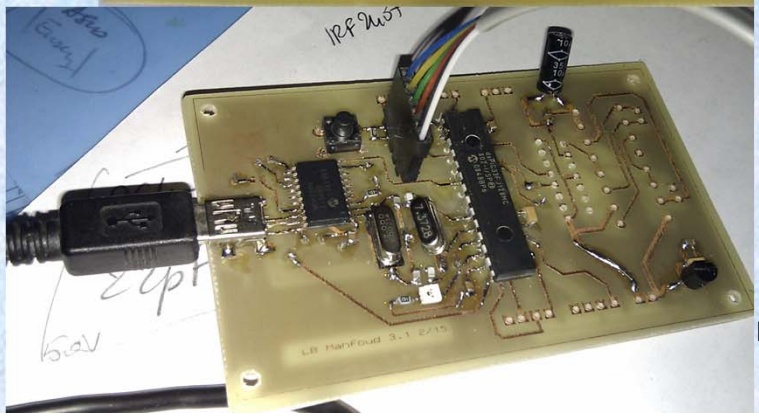


Examples

- dspic 30F3010, LCD 3310, V/f
- PIC 24FJ32GA002, ESP8266
- dspic 33FJ128MC802, MCP2200, L298 : Video Mahfoud 3







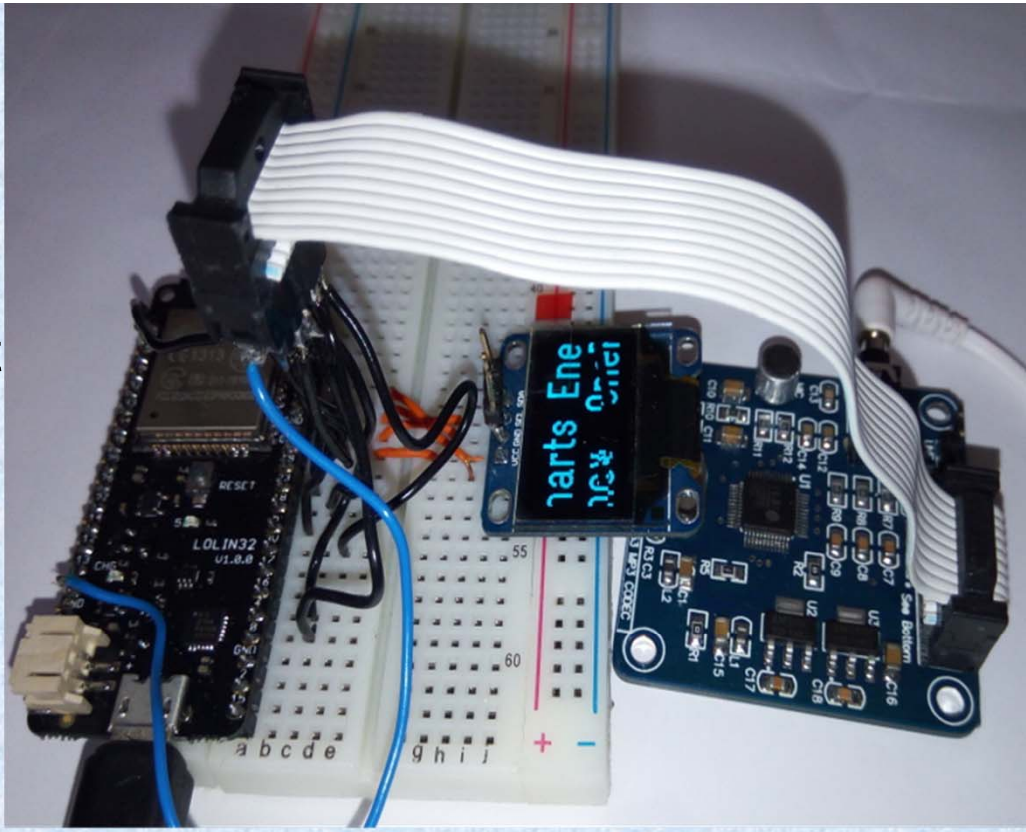
Examples

- Weather station
- Forecast (wunderground.com)
- Internet clock (NTP Client)
- Log measured temperature (Thingspeak Client)
- Customable (github.com/squix78/esp8266-weather-station)

https://github.com/squix78/esp8266-weather-station
https://www.wunderground.com
https://digitalprodsion.com/diy-electronic-weather-station-for-kids-esp8266-internet...

Examples

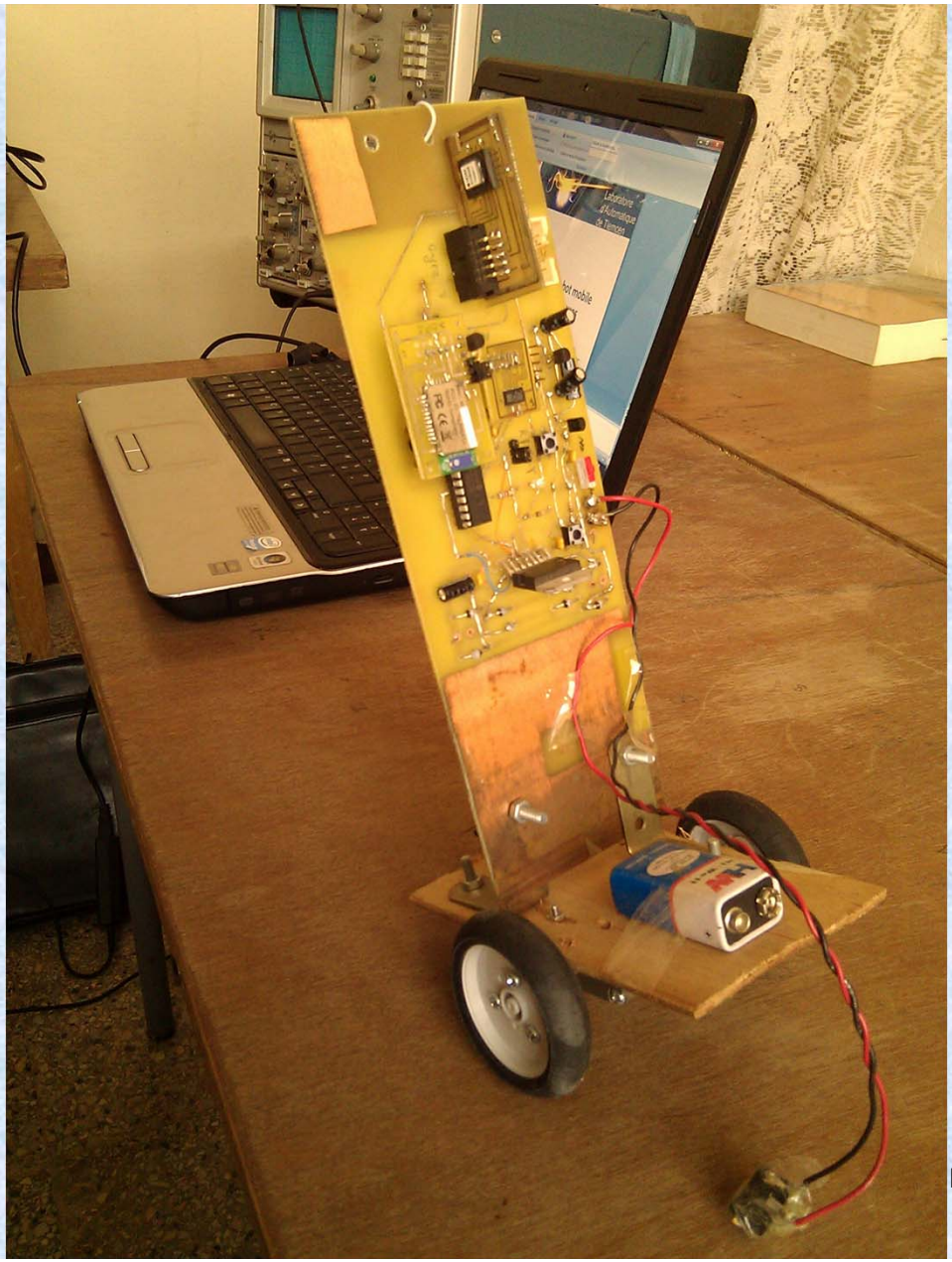
- Webradio
- ESP32 + VS1053
- Connect to Internet radio stations that broadcast MP3 or Ogg audio streams
- Configuration edited through web interface
- Customable)



https://github.com/Edzelf/ESP32-Radio

Examples

- Balancing robot
- dspic 33FJ128MC802
- Chopper 4Q L298
- Gyro AD16250
- Accel LIS3LV02DL
- Bluetooth RN41



dspic Teaching board

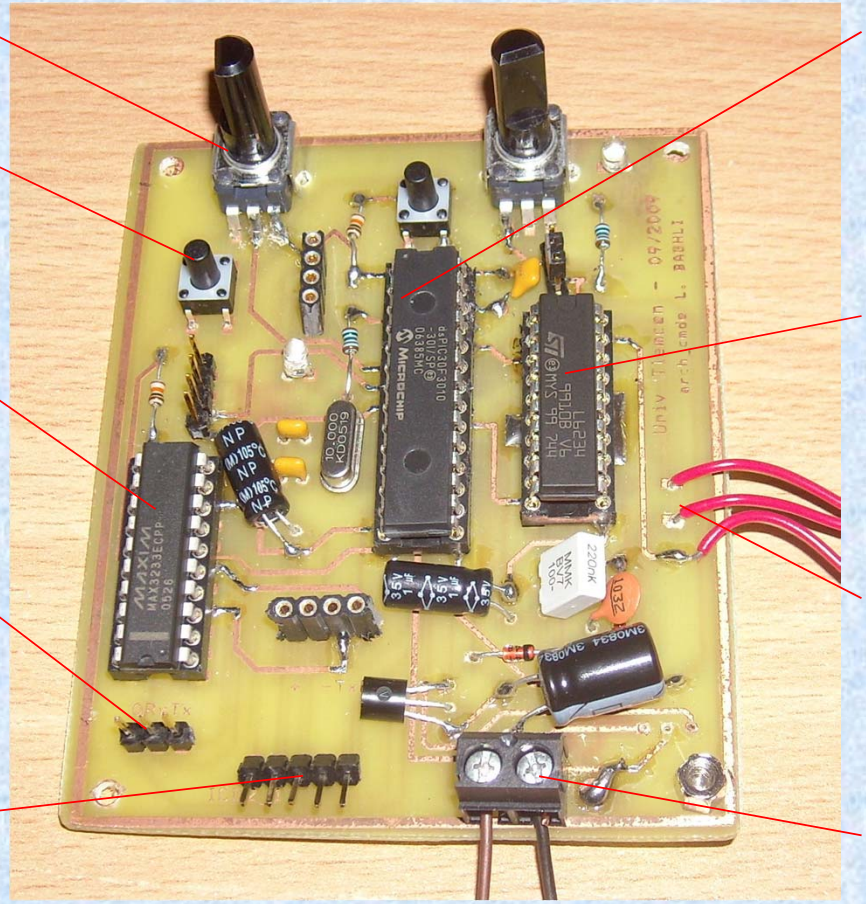
Potentiometer

Push Button

UART – RS232 Interface (MAX3233)

RS232 connecter to PC

PICkit2 programming connecter



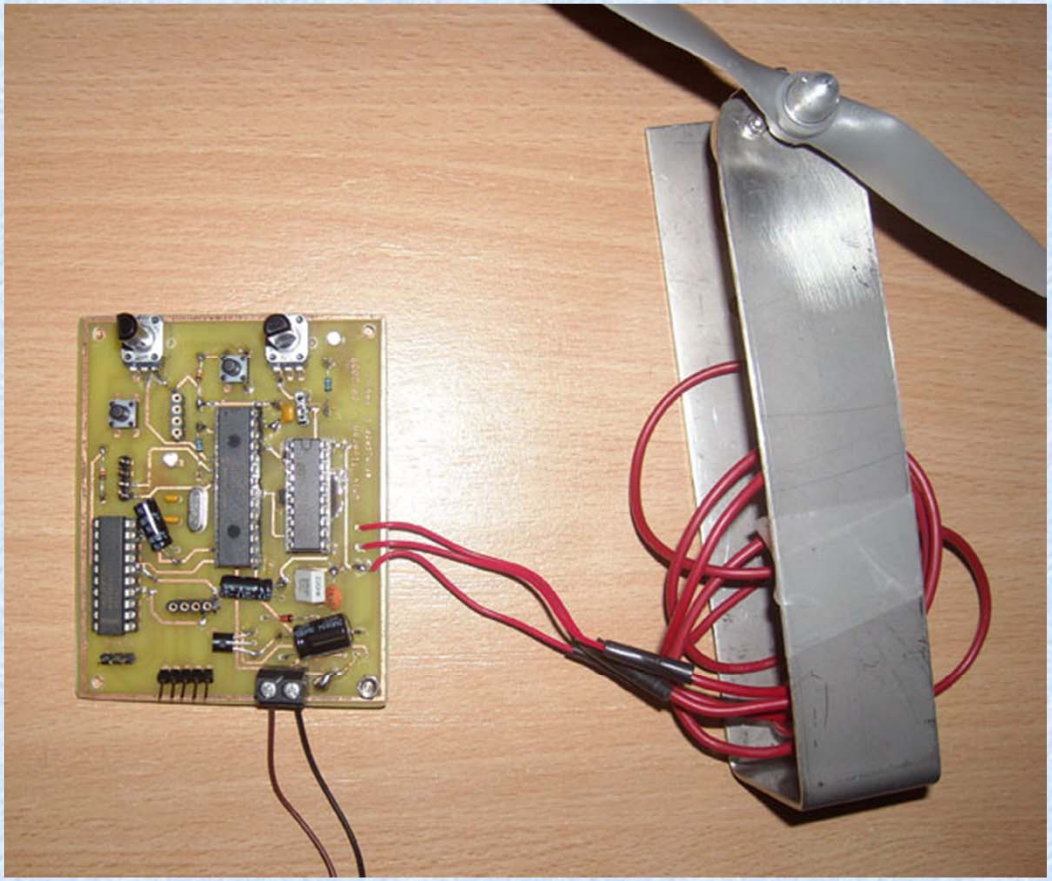
dspic 30F3010

3-phases inverter L6234

3-phases output to a Synchronous Motor

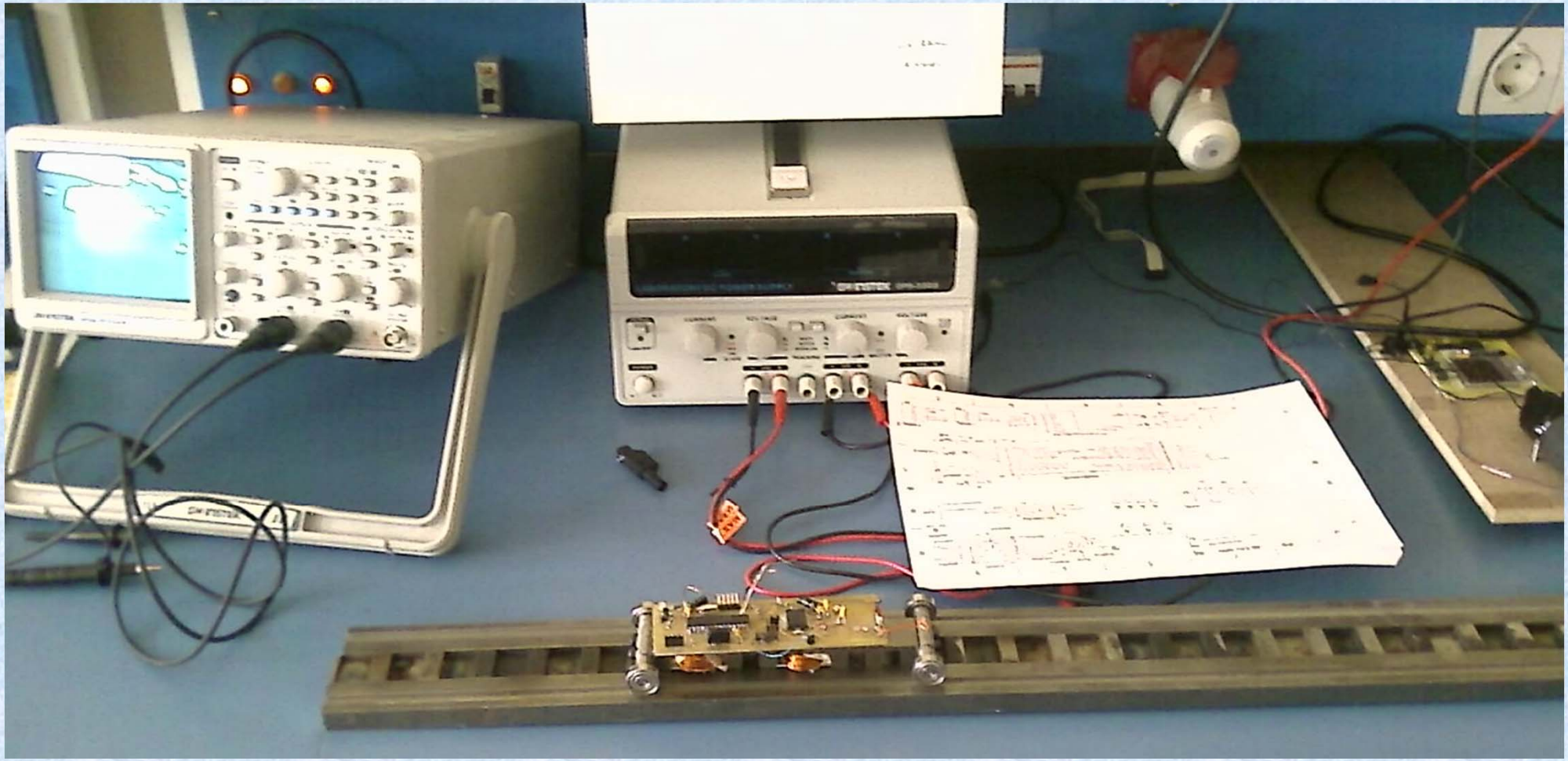
5 to 12 V power input

BLDC (SM) control

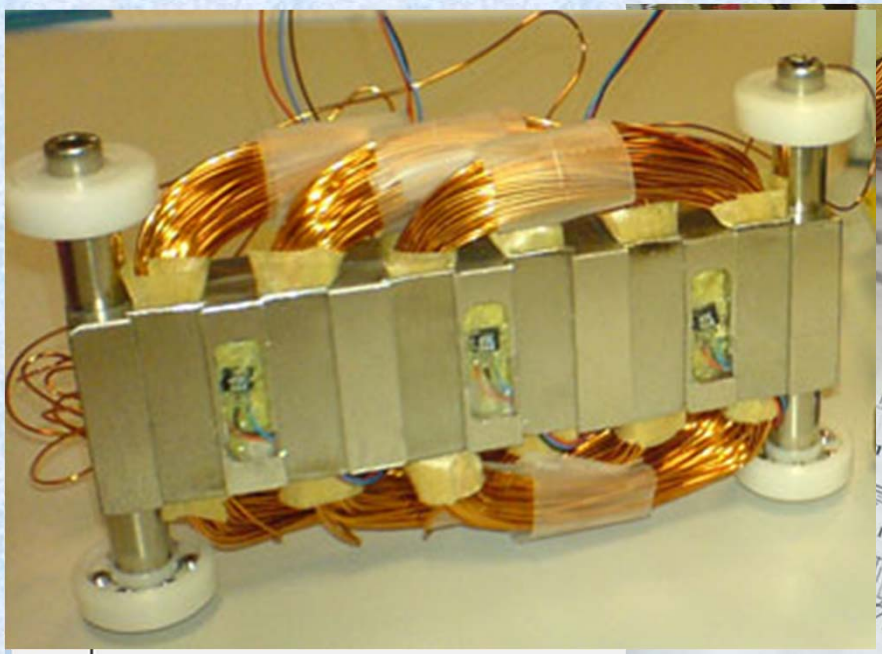


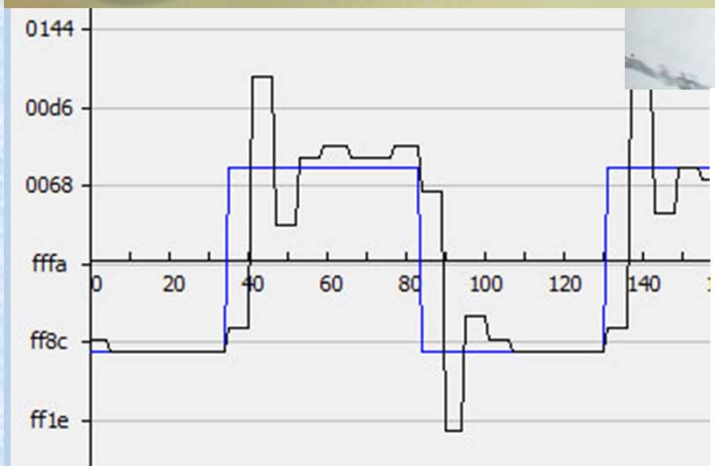
Examples

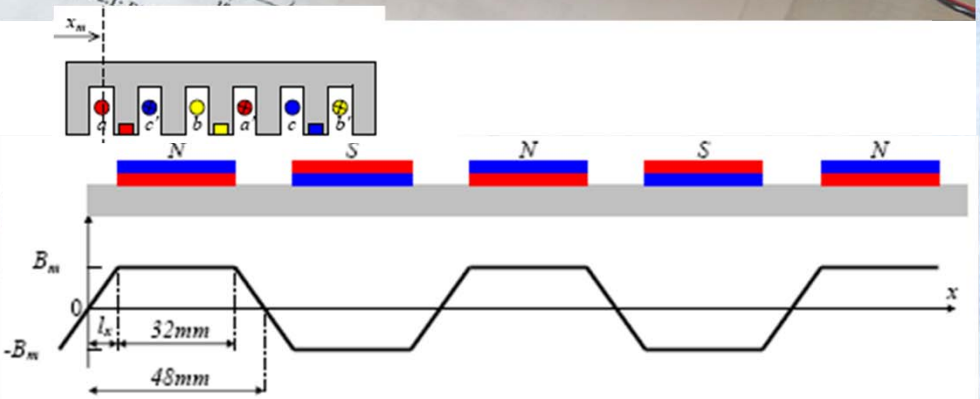
- dspic 33FJ128MC802, Inverter L6234, tri-phase coils



BLDC control

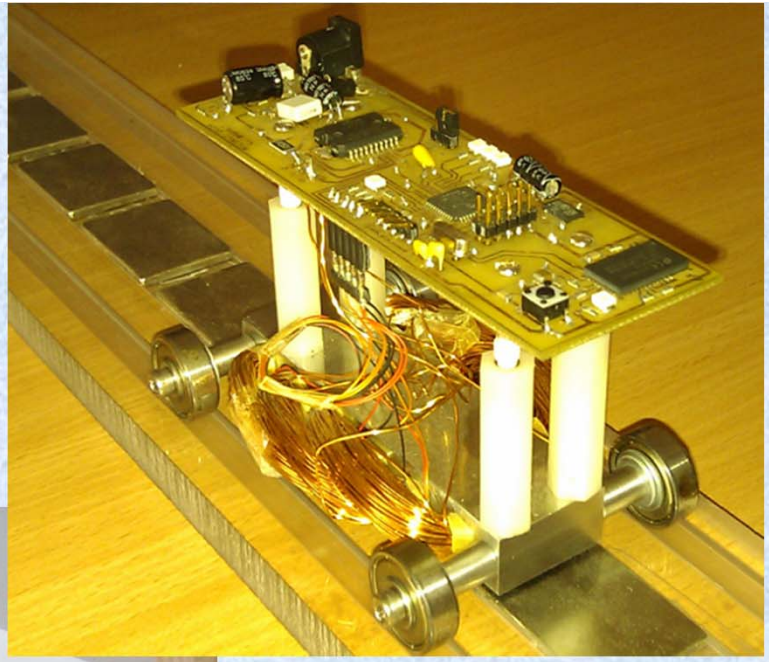






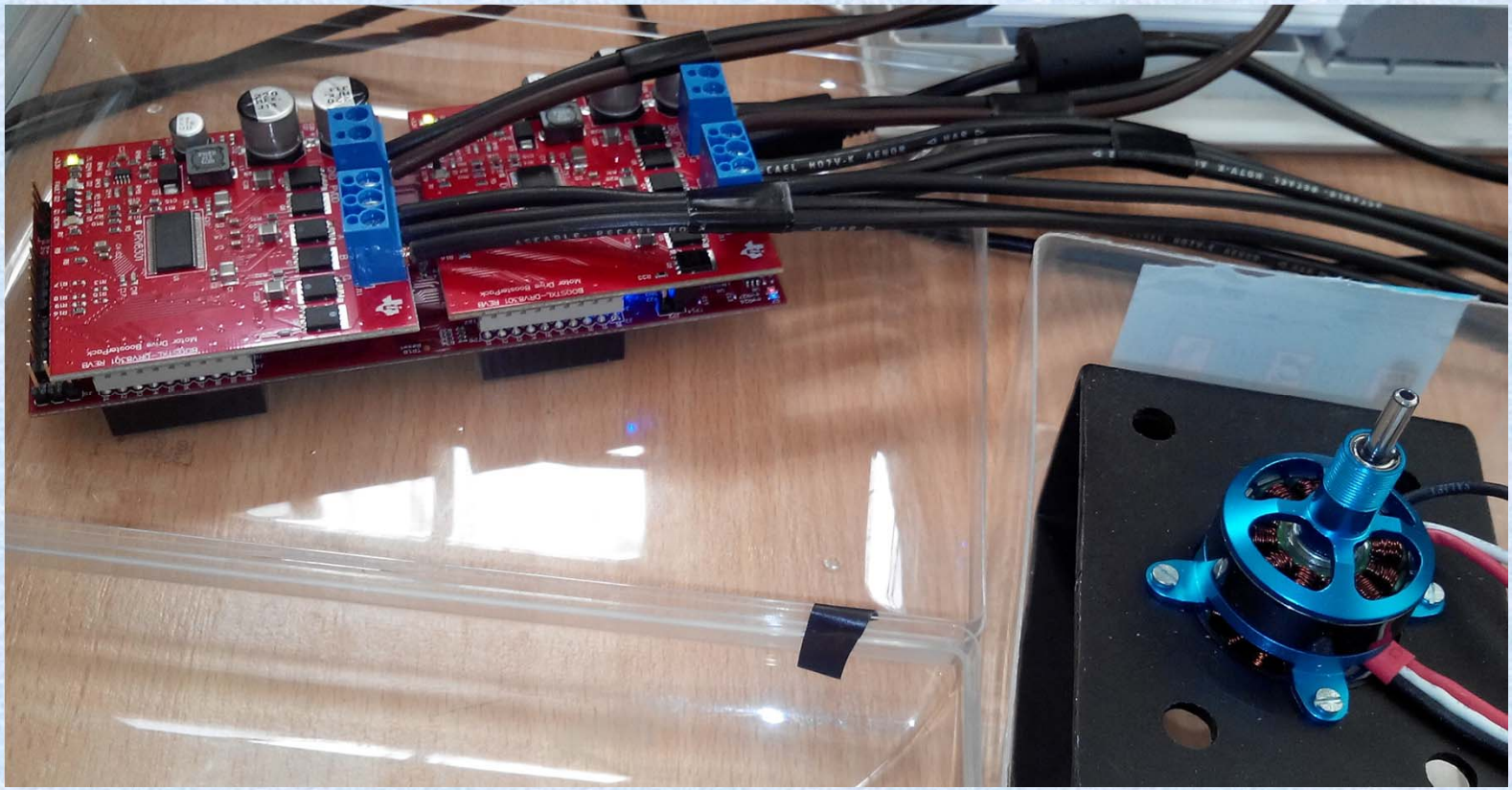
Examples

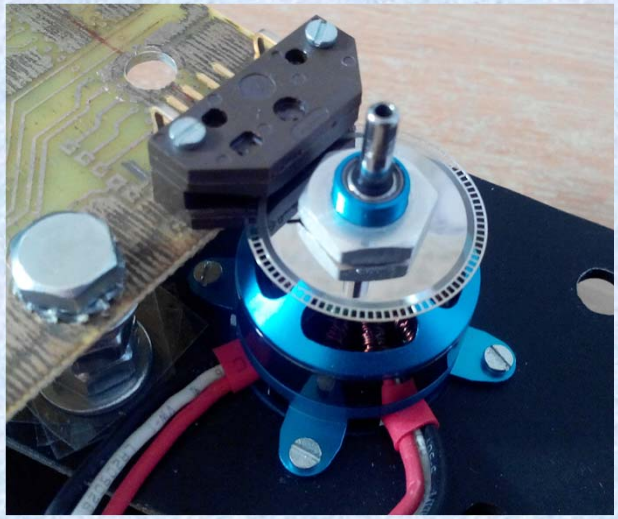
- dspic 33FJ128MC804, Inverter L6234, Bluetooth LMX9838, Accel LIS3LV02DL, Hall SS40A, tri-phase coils: Video

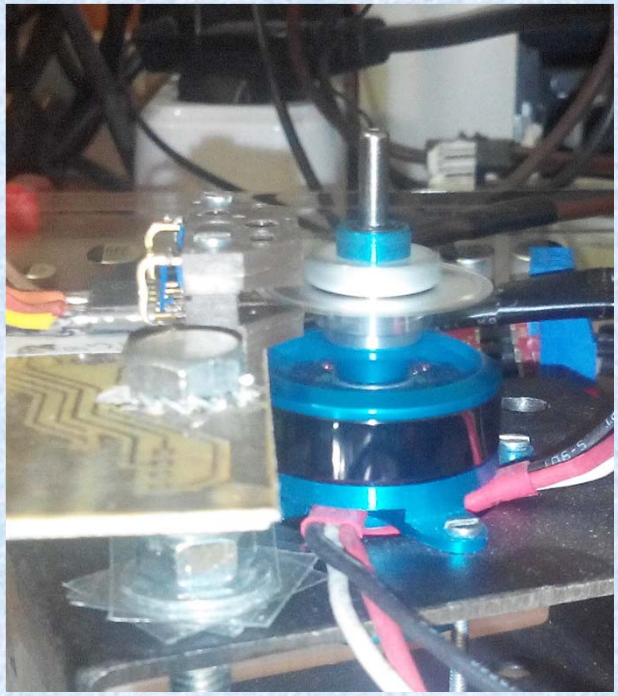


Examples

- LaunchpadXL F28069M, incremental encoder, 2 x DRV8301, BLDC motors







Energy transfer and motor control

Requirements:

- Power Electronics: Inverter, DC chopper, rectifier
- Motors (generators): IM, DFIG, PMSM, ~~DC motor~~
- Energy harvesting: PV, Wind turbine
- Encoder, current and voltage sensors, PWM interface
- Control board, interfaces
- Control methods: Vector control, V/f, BLDC, PWM, MPPT
- Control programs; run, log and supervise

Control boards for research

Advantages:

- Powerful boards
- Easy to use
- Compatible with industry R&D
- Reusable code (Matlab / Simulink)

Disadvantages:

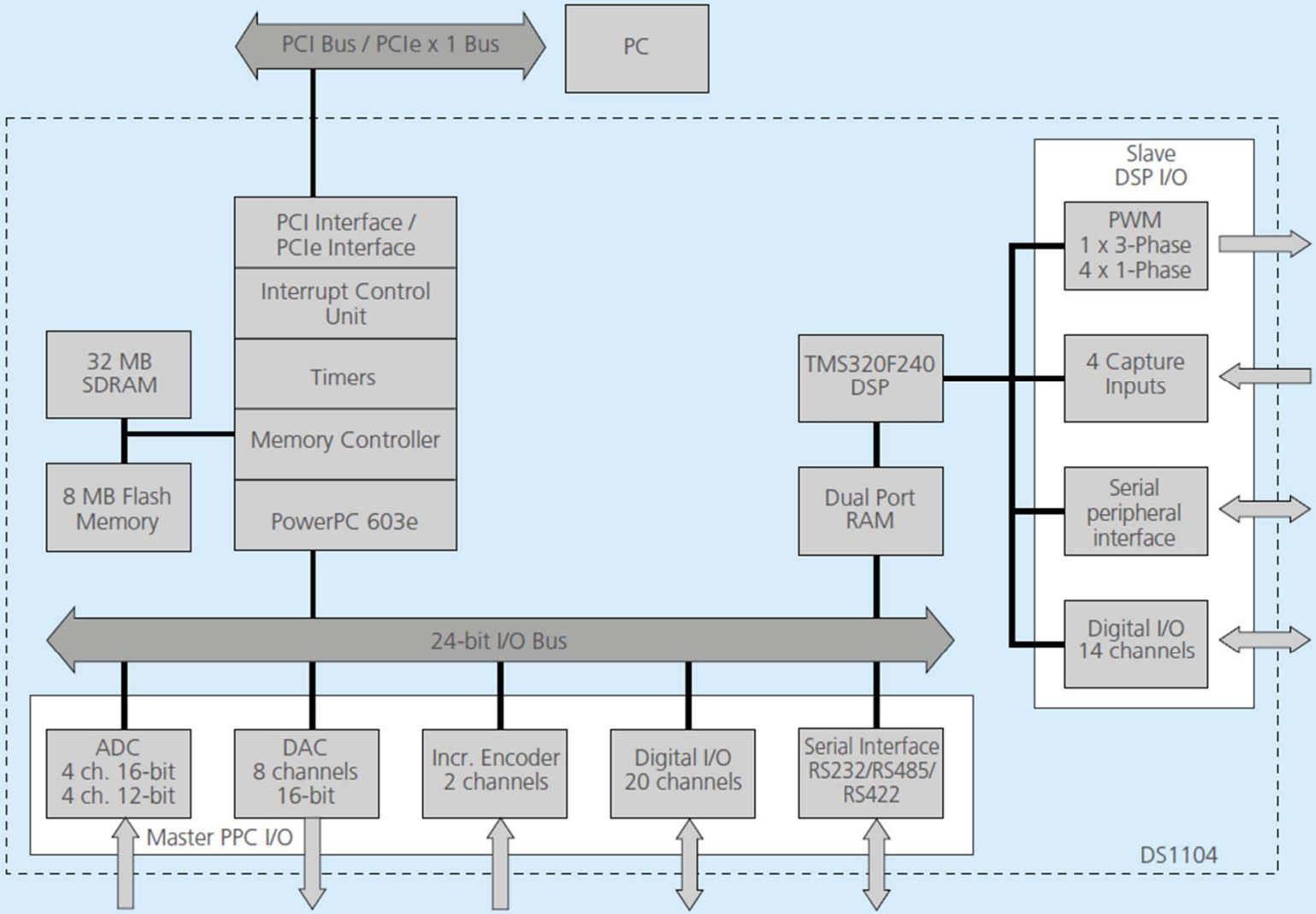
- Expensive
- Difficult to adapt the study to the industrial world

dSpace offers ACE Advanced Control Education Kit

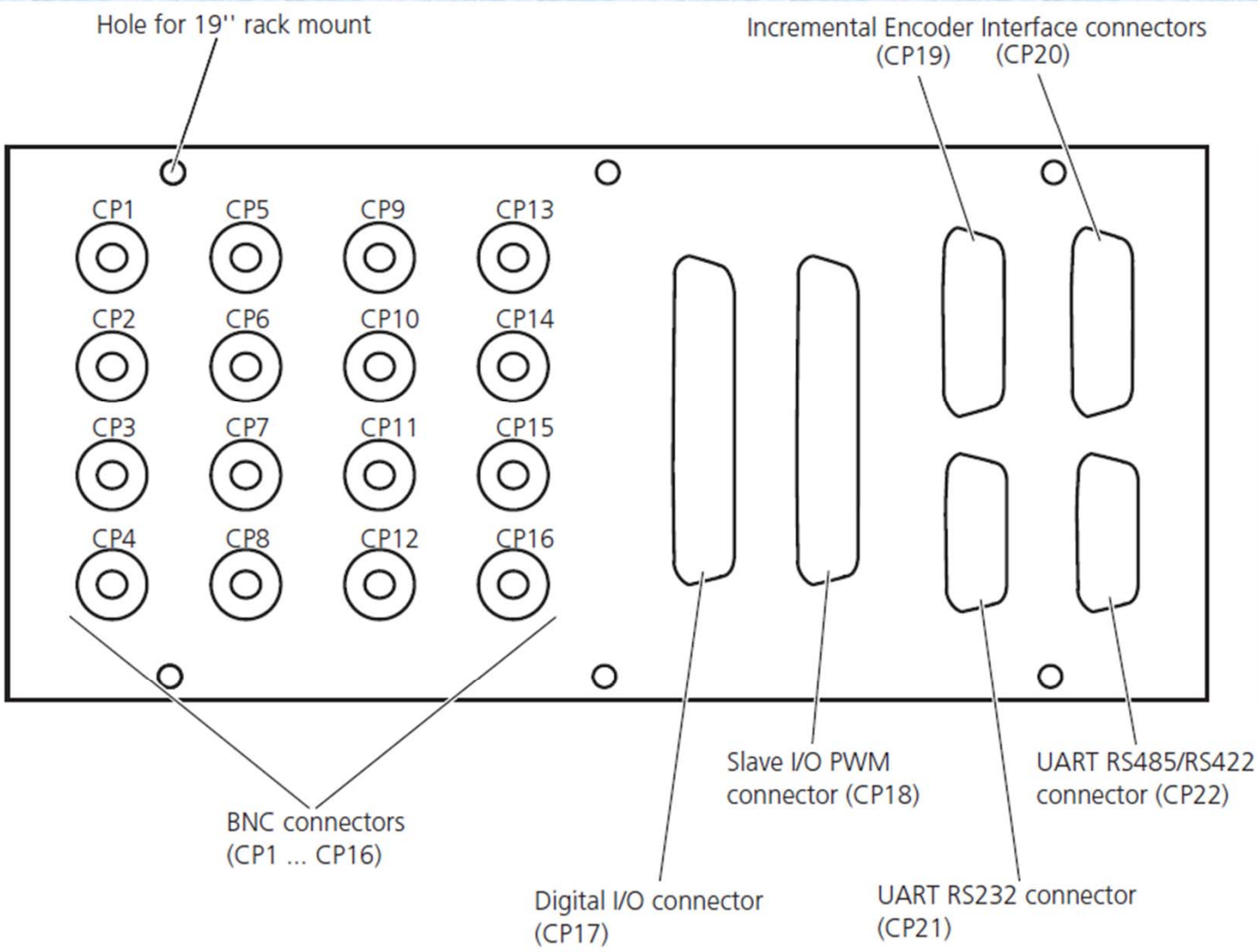
- DS1103: ISA bus or PX4-Ethernet/PCI powerful but retired (10 k€)
- DS1104: PCI bus; best offer (5 k€)
- MicroLabBox : Ethernet; laboratory(11 k€)
- MicroAutobox II : Ethernet; embedded (16 k€)

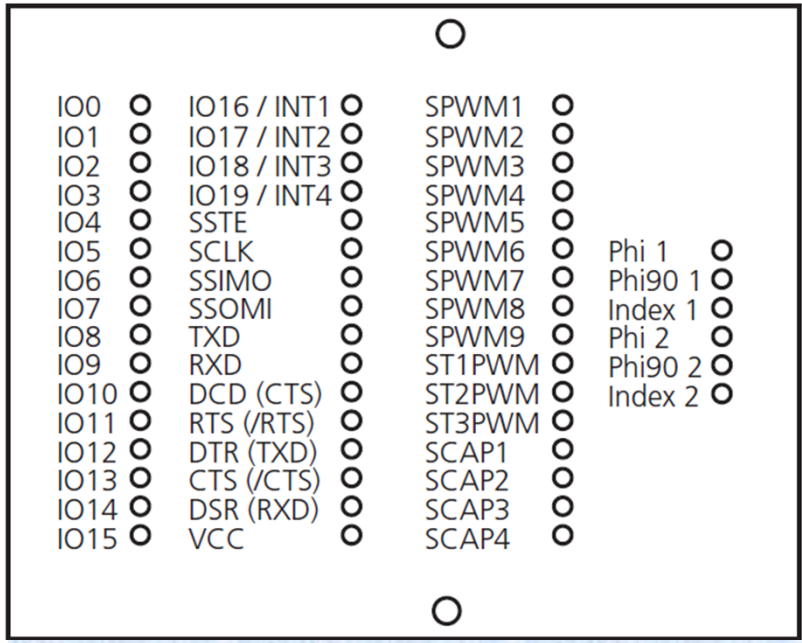
https://www.dspace.com/fr/fra/home/products/hw/singbord/ds1104.cfm
https://www.dspace.com/fr/fra/home/products/hw/micautob.cfm
https://www.dspace.com/fr/fra/home/products/hw/microlabbox.cfm

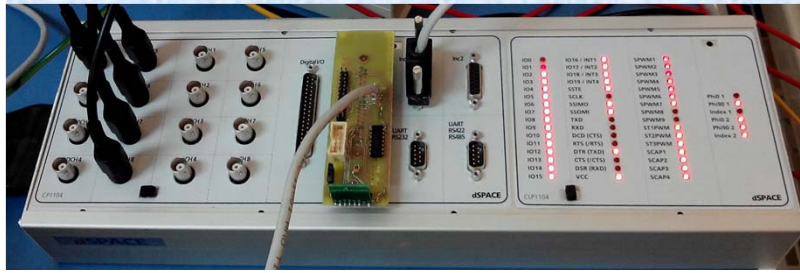
DS1104 board



Extension board CLP1104



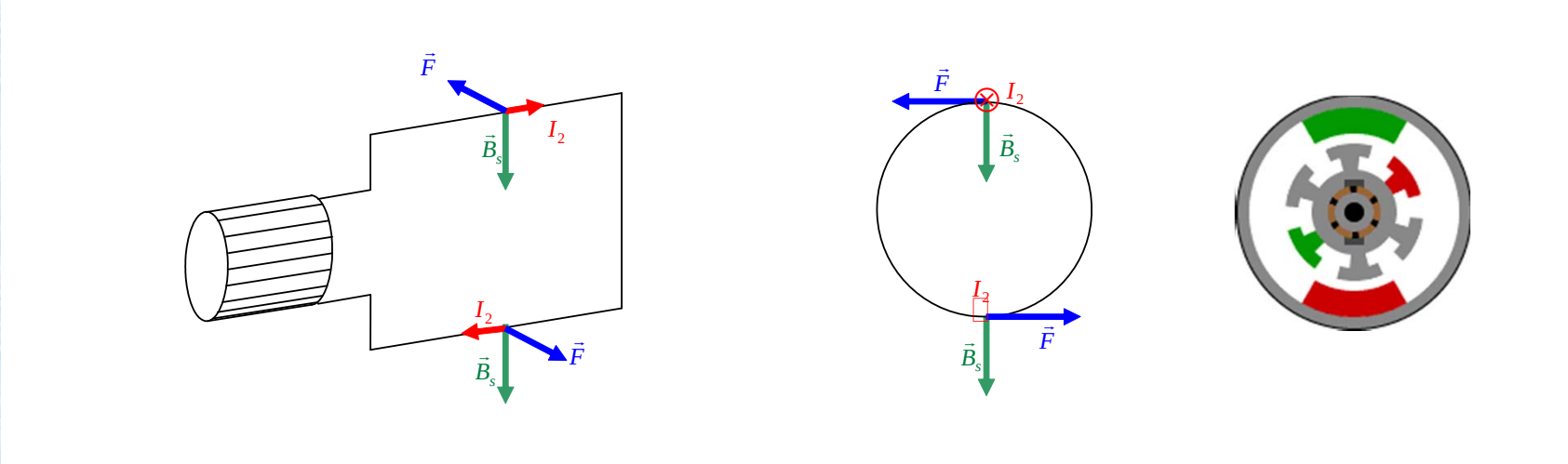




Motors

Design and operating Principle

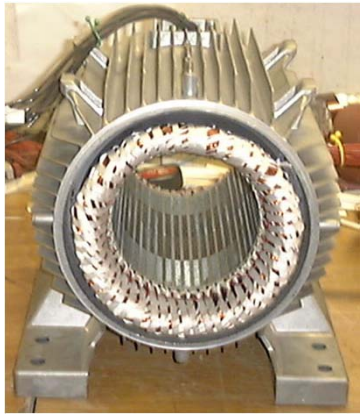
- DC motor
  - System: commutator, brushes
  - Stator flux: External current or PM (Permanent Magnets)

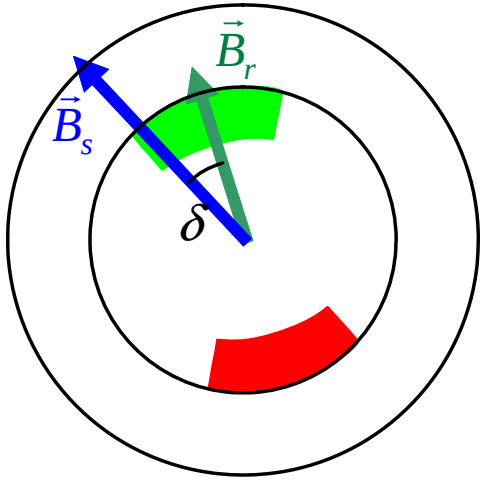


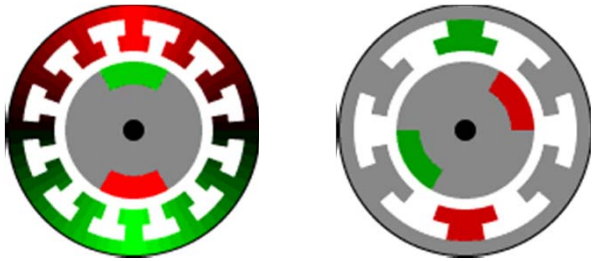
Motors

Design and operating Principle

- Synchronous Motor (SM)
  - 3-phases stator, rotating field
  - Rotor à External current or PM (Permanent Magnets)



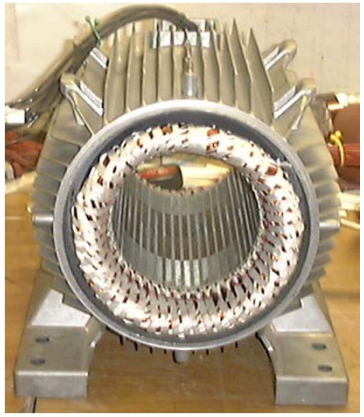


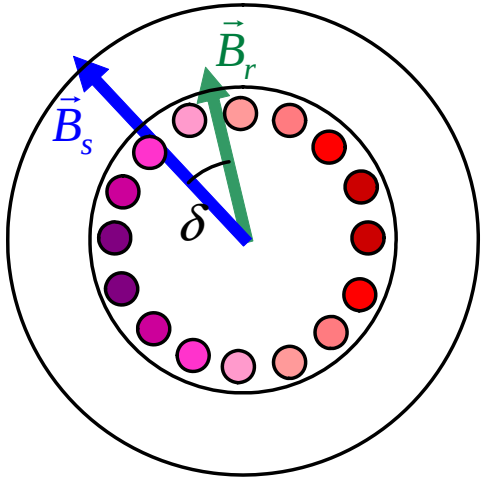


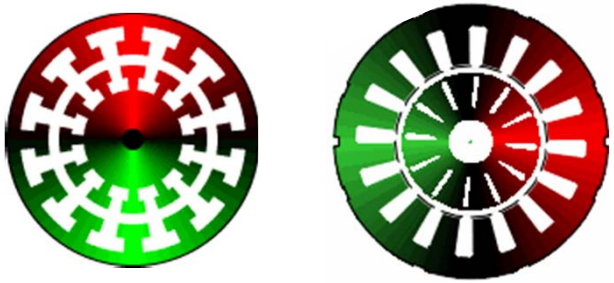
Motors

Design and operating Principle

- Induction Motor (IM)
  - Same stator as SM
  - Rotor: Squirrel cage or wound rotor, slip



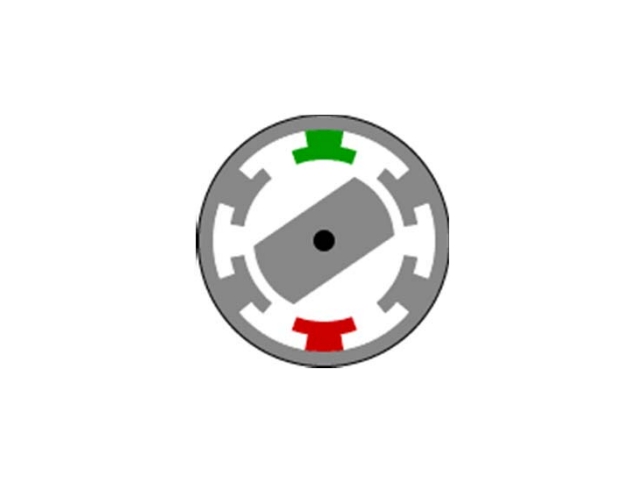




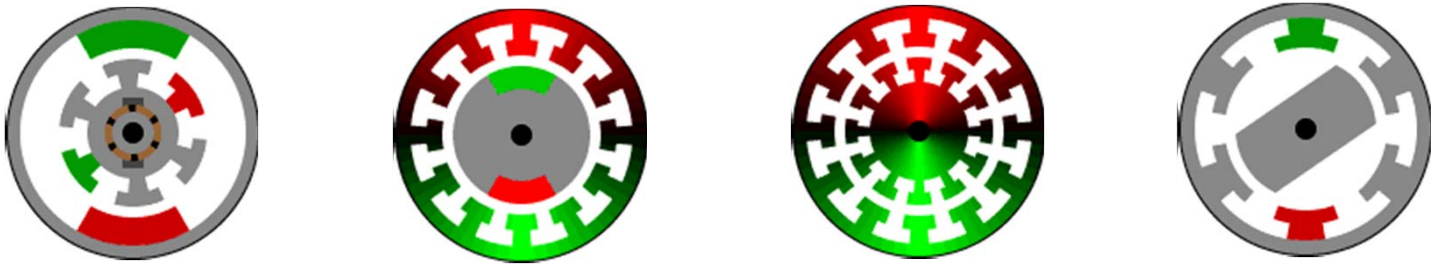
Motors

Design and operating Principle

- SRM
  - N-phases stator
  - Reluctant rotor (no coils)
  - Principle of minimum magnetic circuit reluctance



Motor comparison



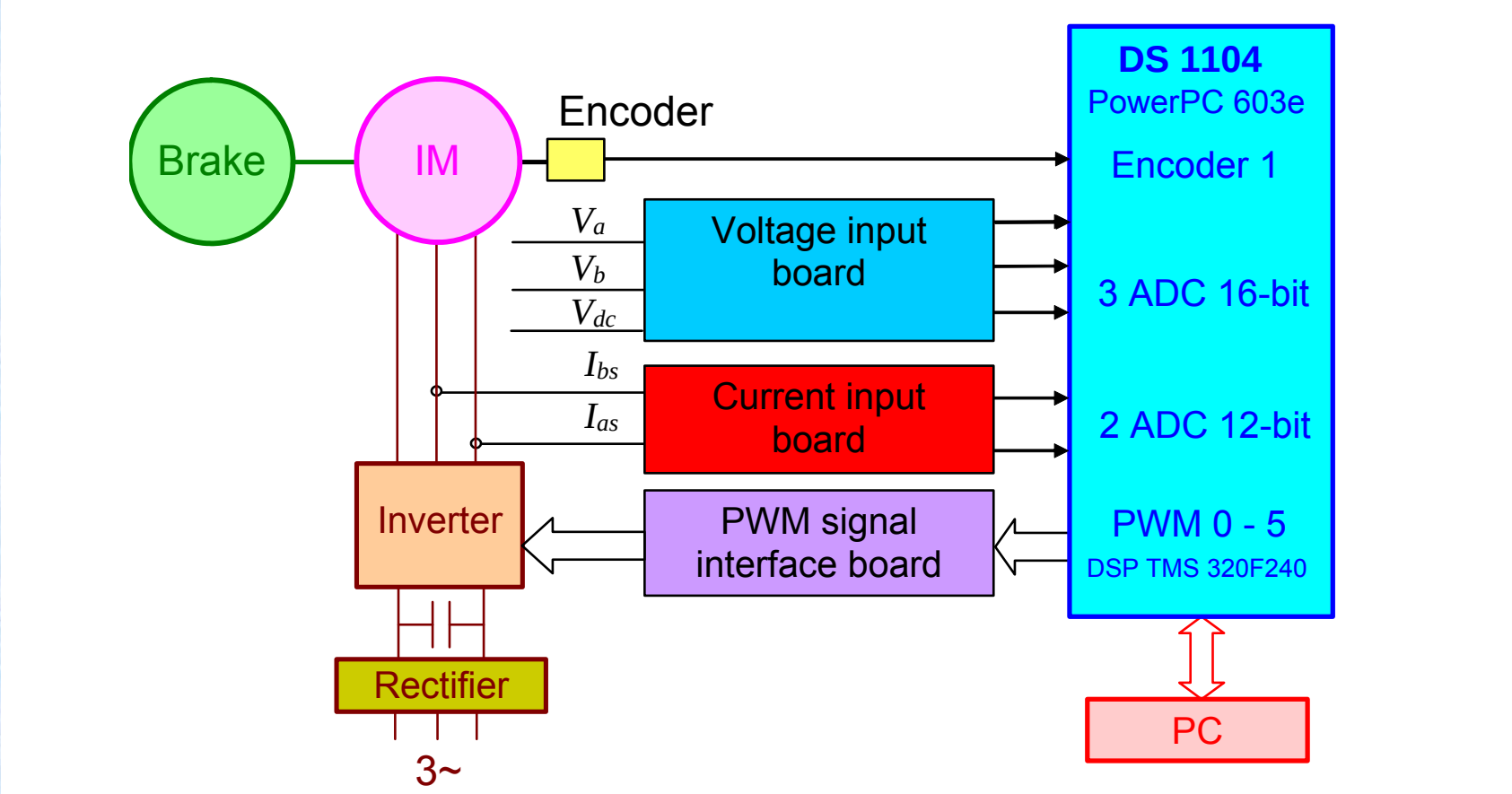
DC SM IM SRM

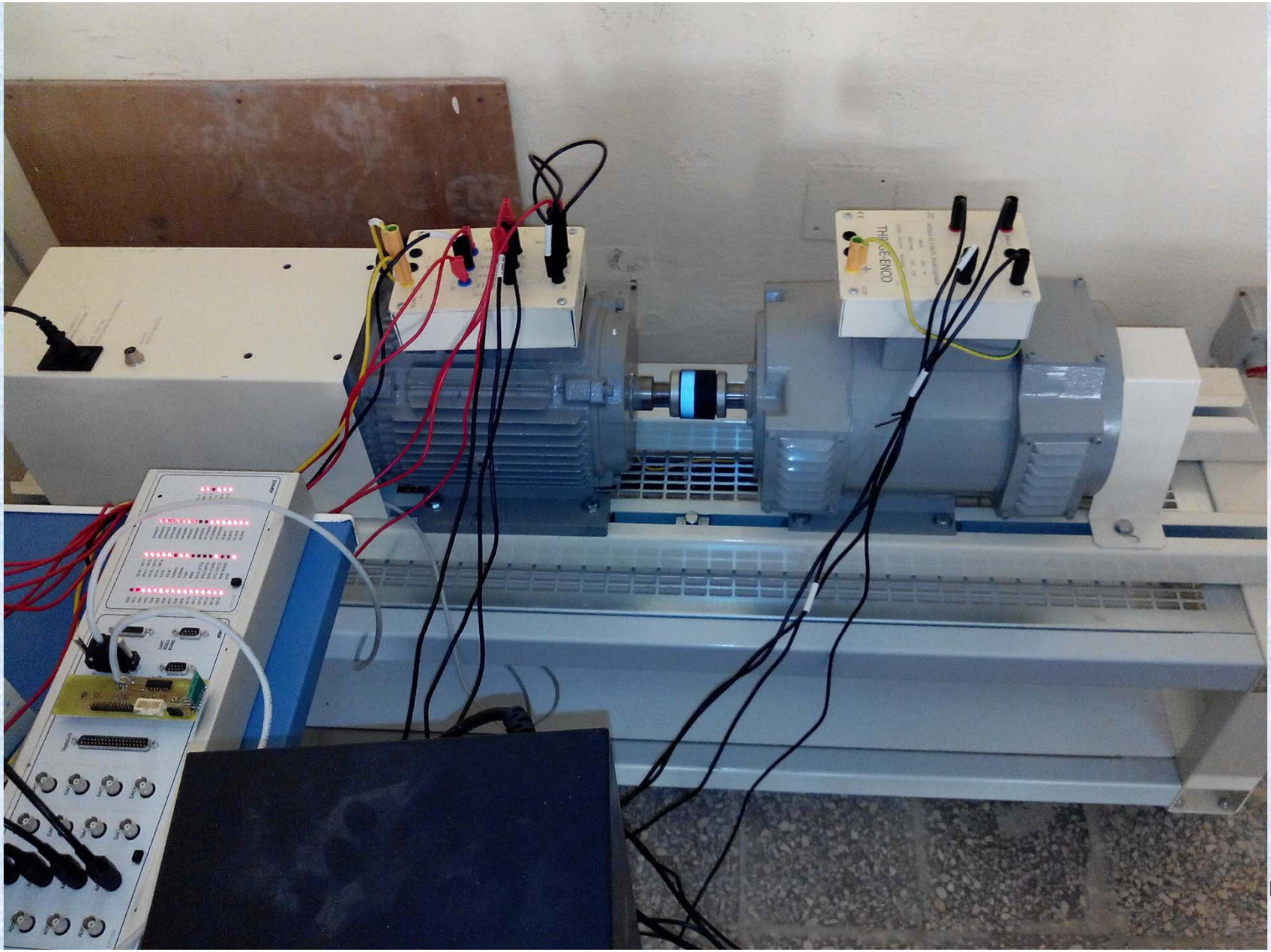
Type	Simplicity	Power Electronics	Reliability	Cost	Mecanical power/Torque
DC	+	+	-	+	-
SM	-	0	-	+	+
IM	-	0	+	-	-
SRM	-	-	+	-	-

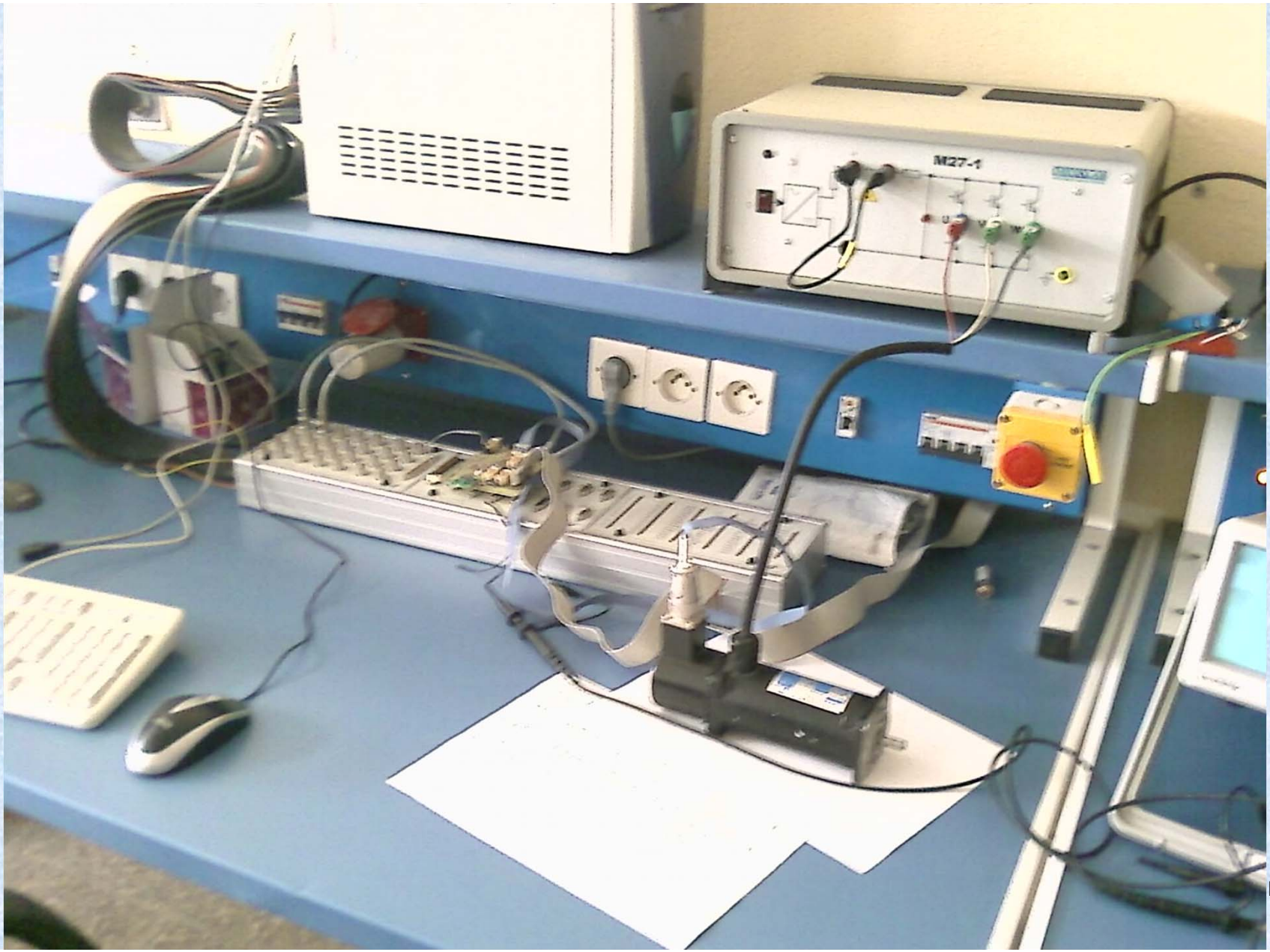
Motor control

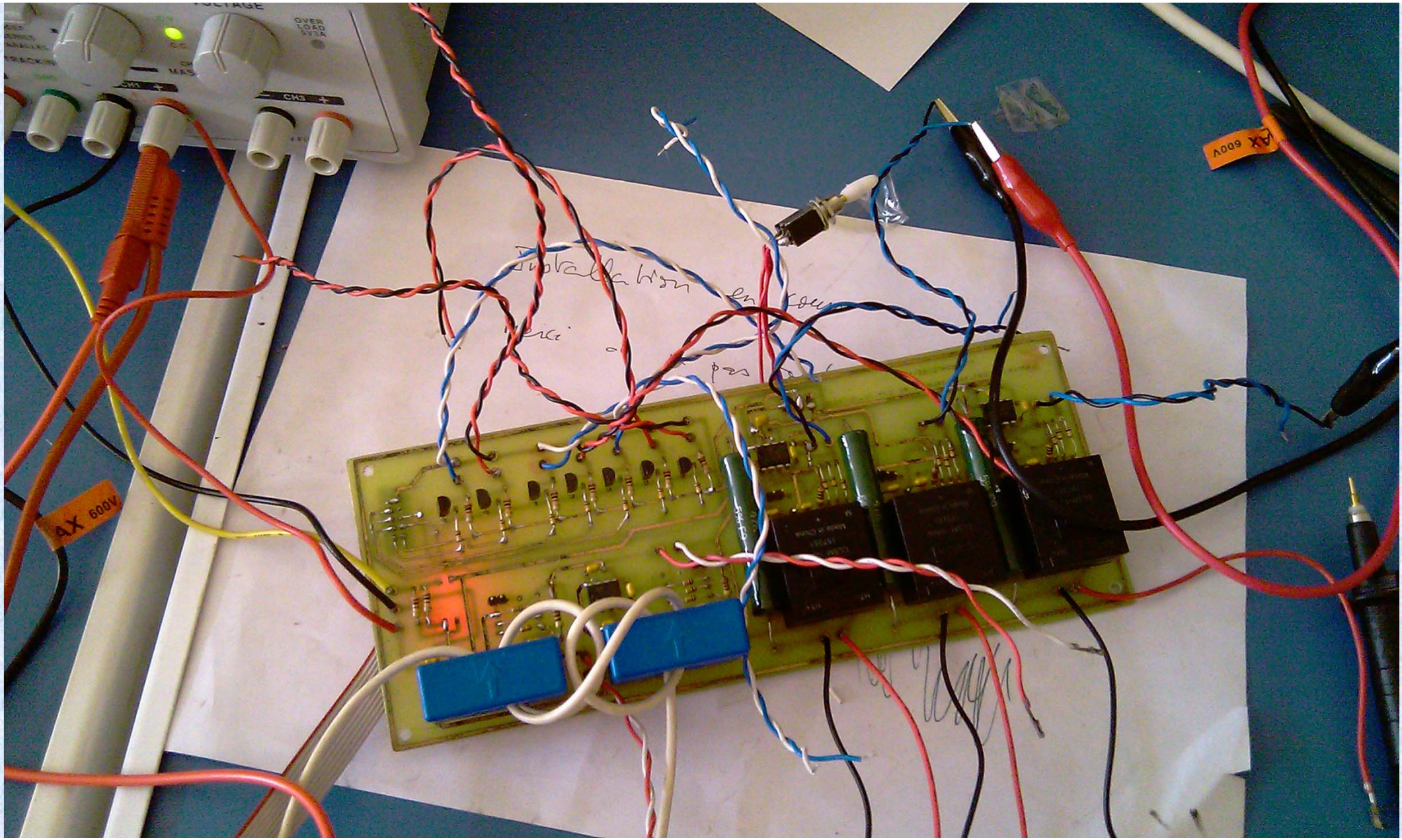
- DC motor
  - Simple, decoupled system
  - Analog or digital control of a DC chopper
- SRM
  - Powere phase by phase
  - Need to know rotor position
- SM, IM
  - V/f (in industry) not in power drive and transportation
  - Vector control: IRFO, DSFO,… DTC
  - Vector control ➢ incremental / absolute encoder / resolver, sensorless
  - BLDC (SM) ➢ Hall effet sensors, sensorless

IRFO Control of Induction Machine

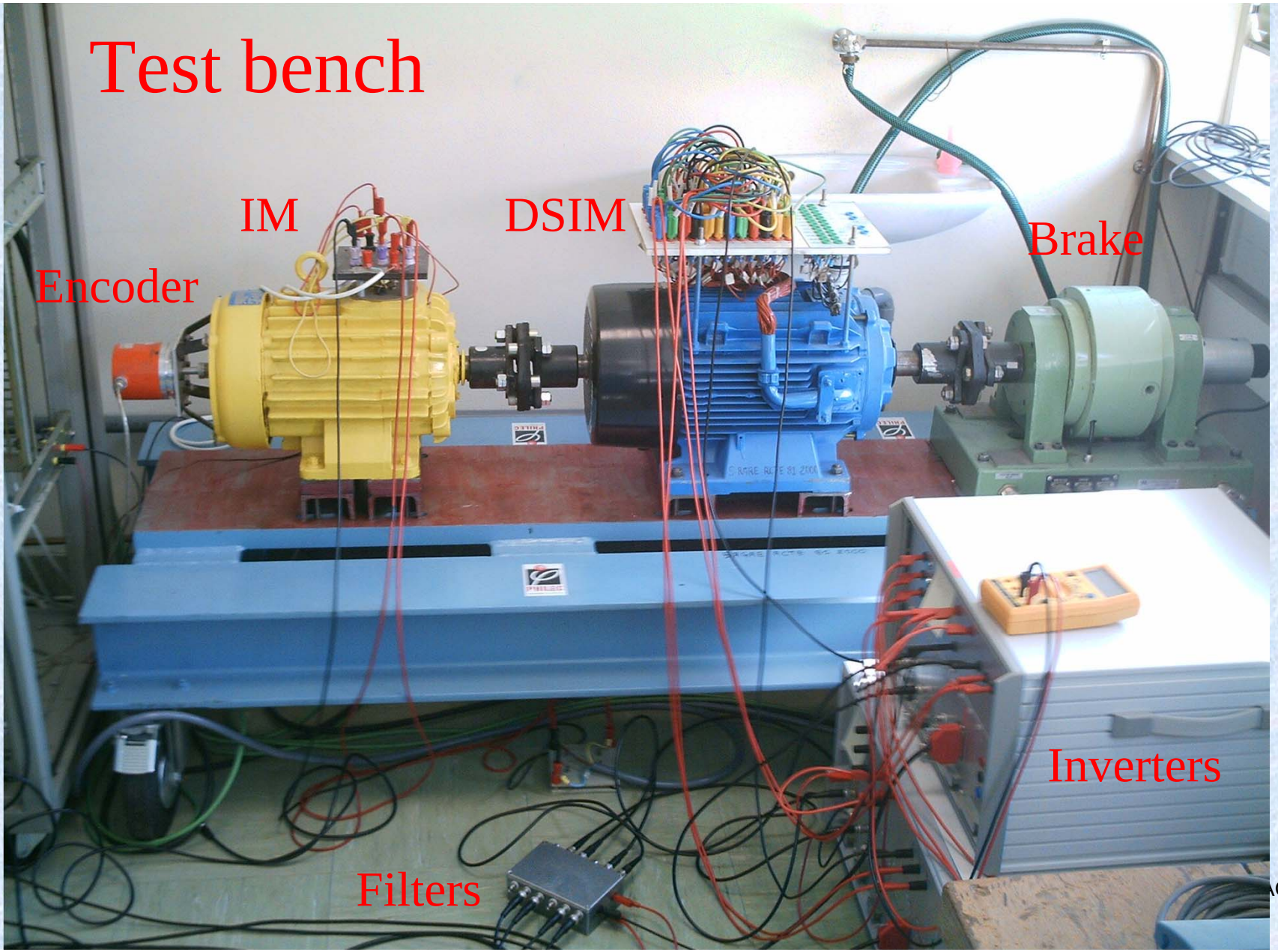



THR-CE-ENCO


M27-1

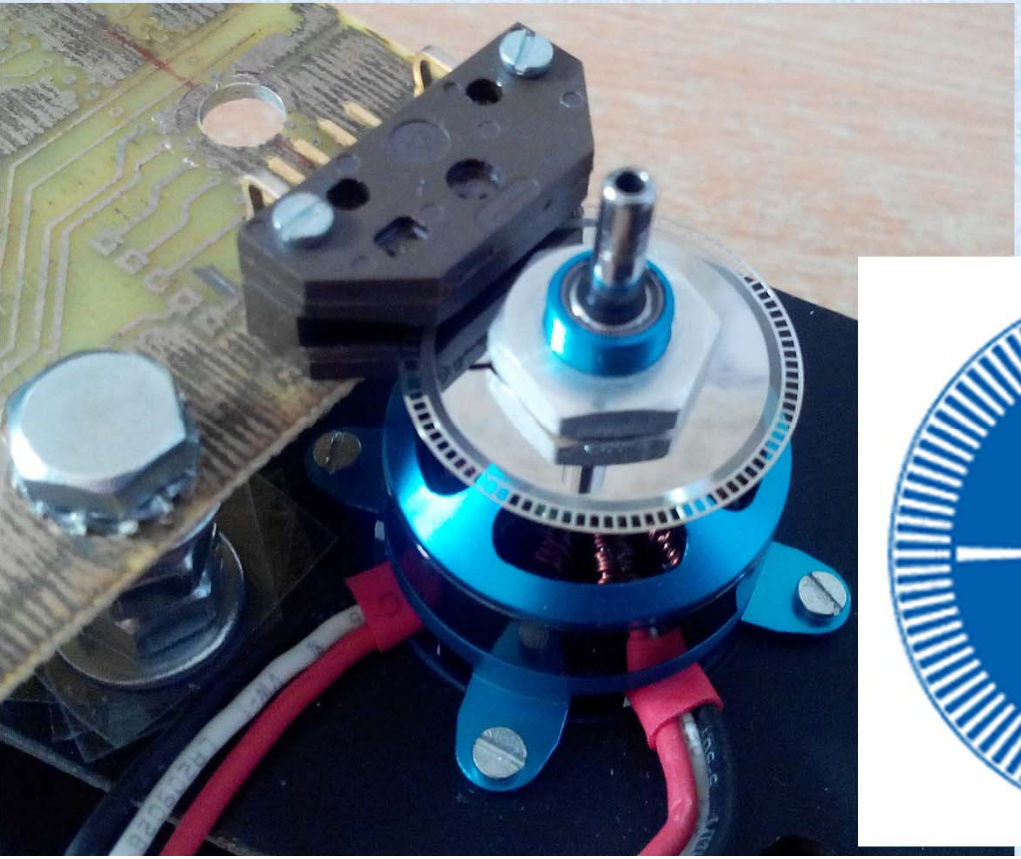

VOLTAGE
OVER LOAD
CH1
CH2
4X 600V

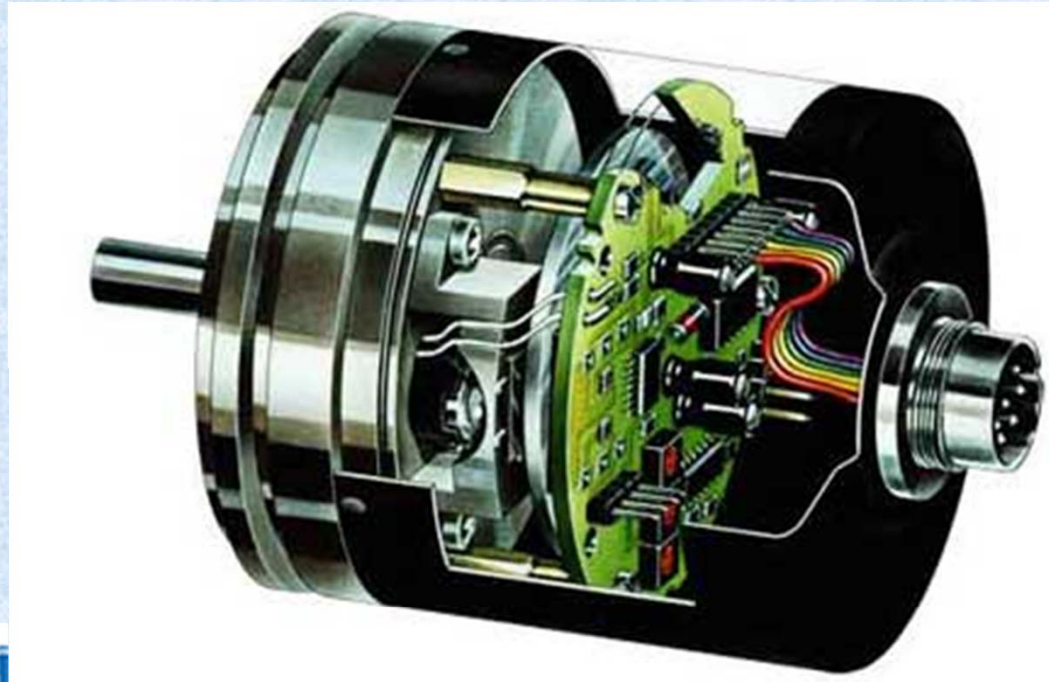
Test bench

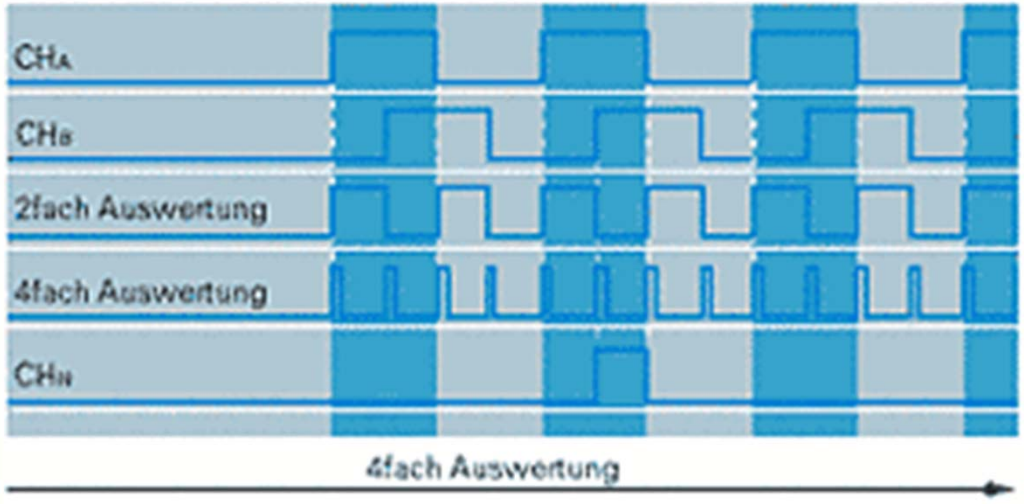


Incremental encoder

- Technology optoelectronic
- Digital output

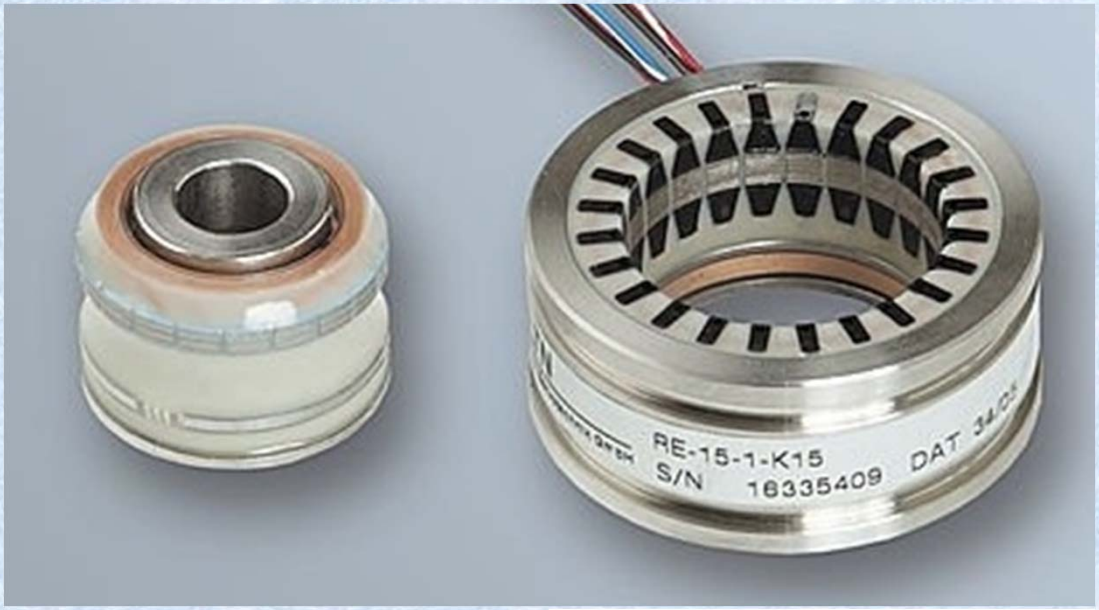


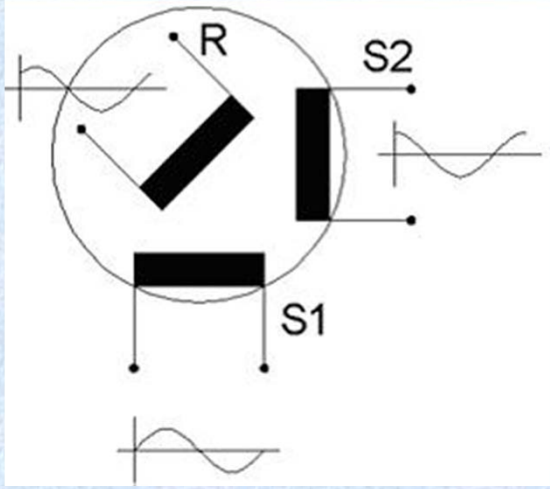




Other encoders

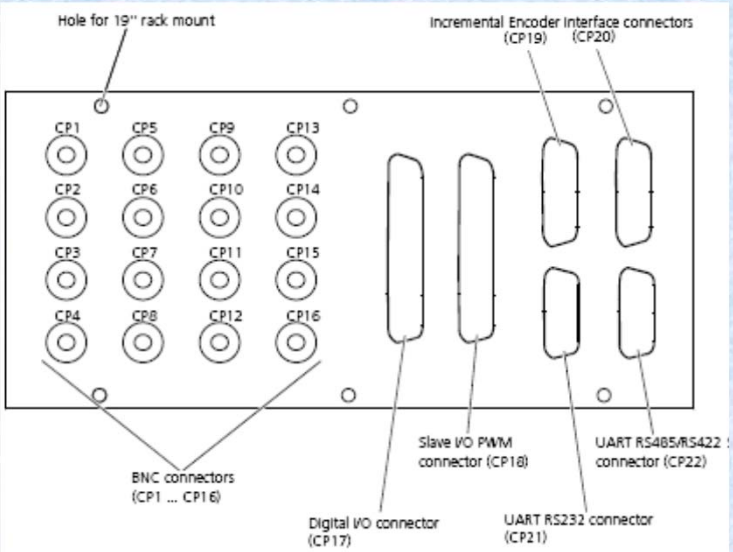
- Absolute encoder
- Resolver
- for SM



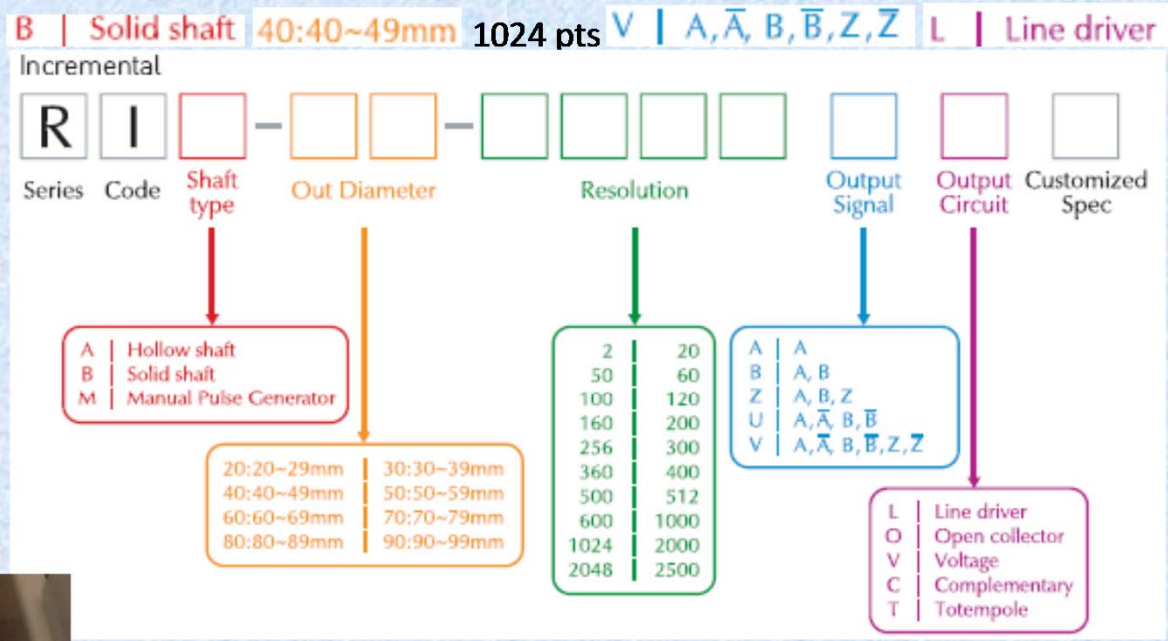


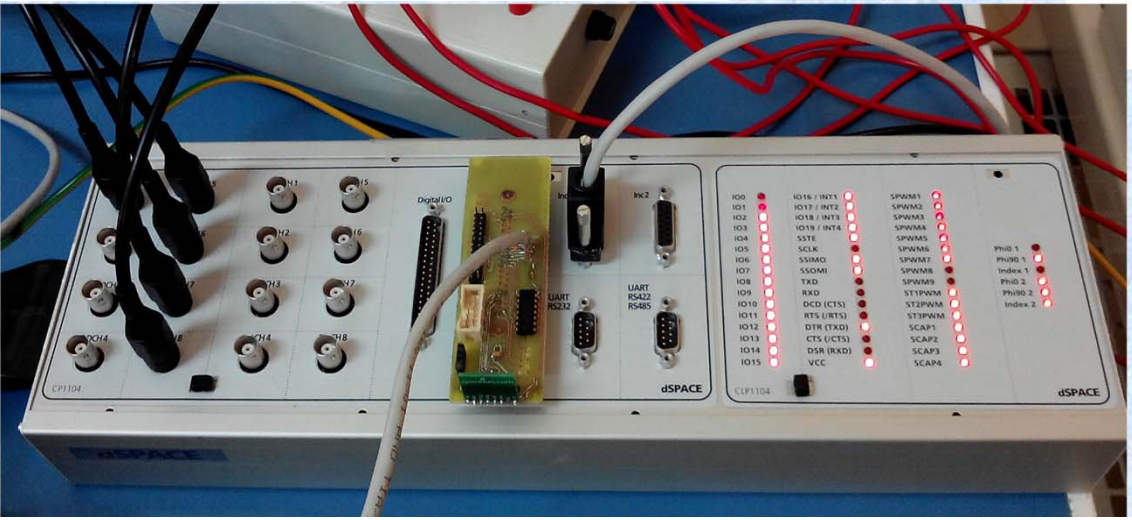
Incremental encoder

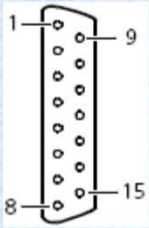
Langlois **RIA 40 1024 V L**



connation
To CLP 1104

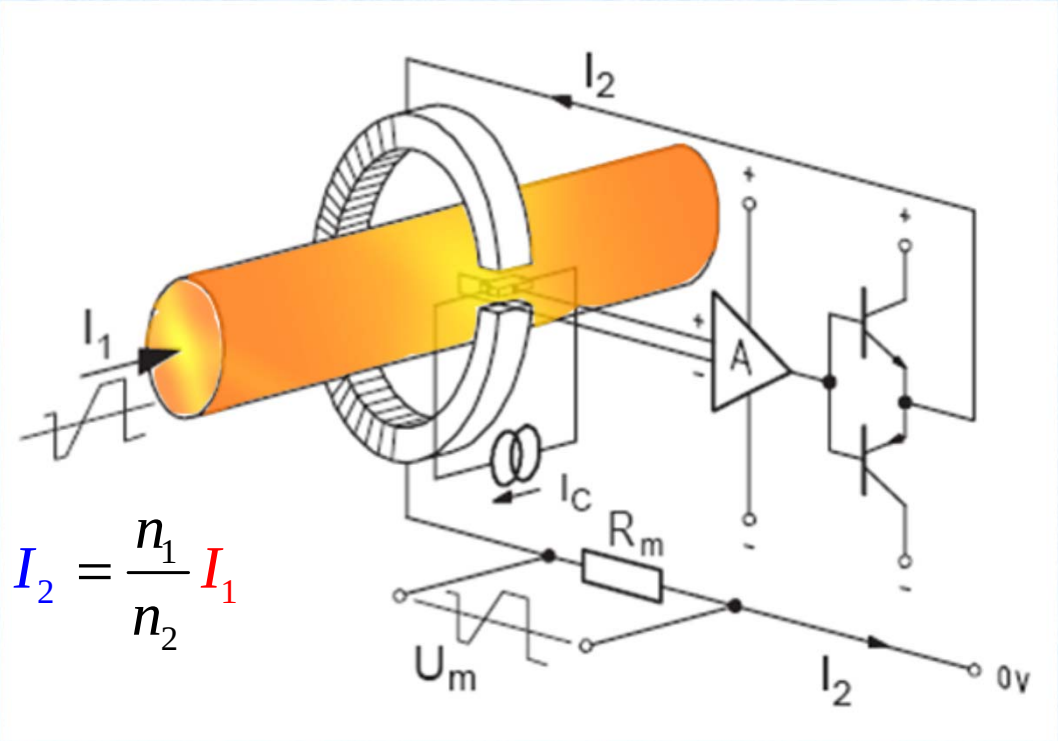




Connector (CP19, CP20)	Pin	Signal	Câble Ethernet	Pin	Signal
	1	VCC (+5 V)	Marron		
	2	PHI0(x)	vert	9	VCC (+5 V)
	3	PHI0(x)	blanc-vert	10	GND
	4	PHI90(x)	Bleu	11	GND
	5	PHI90(x)	blanc-bleu	12	GND
	6	IDX(x)	Orange	13	GND
	7	IDX(x)	Blanc Orange	14	GND
	8	GND	blanc-marron	15	GND

Current sensors

- Hall effect technology
- Galvanic isolation
- Measures in:
  - Continuous
  - Alternate
  - Instantaneous
- I → V
- Filtering
- ADC



$$I_2 = \frac{n_1}{n_2} I_1$$

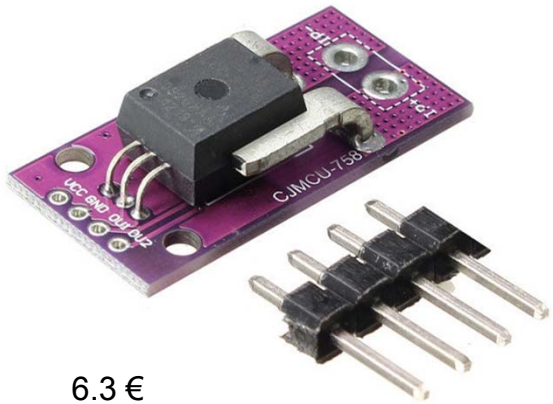
Current sensors

Hall effect current sensors

Allegro ACS 758

- Industry-leading noise performance through proprietary amplifier and filter design techniques
- Integrated shield greatly reduces capacitive coupling from current conductor to die due to high dV/dt signals, and prevents offset drift in high-side, high voltage applications
- Total output error improvement through gain and offset trim over temperature
- Small package size, with easy mounting capability
- Monolithic Hall IC for high reliability
- Ultra-low power loss: 100 μΩ internal conductor resistance
- Galvanic isolation allows use in economical, high-side current sensing in high voltage systems
- 3.0 to 5.5 V, single supply operation
- 50A working current, -40℃ ~ +150 ℃ working temperature
- 120 kHz typical bandwidth
- 3 μs output rise time in response to step input current
- Output voltage proportional to AC or DC currents
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis

http://www.allegromicro.com/en/Products/Current-Sensor-ICs/Fifty-To-Two-Hundred-Amp-Integrated-Conductor-Sensor-ICs.aspx
http://www.allegromicro.com/en/Products/Current-Sensor-ICs/Zero-To-Fifty-Amp-Integrated-Conductor-Sensor-ICs.aspx



6.3 €

ACS 758

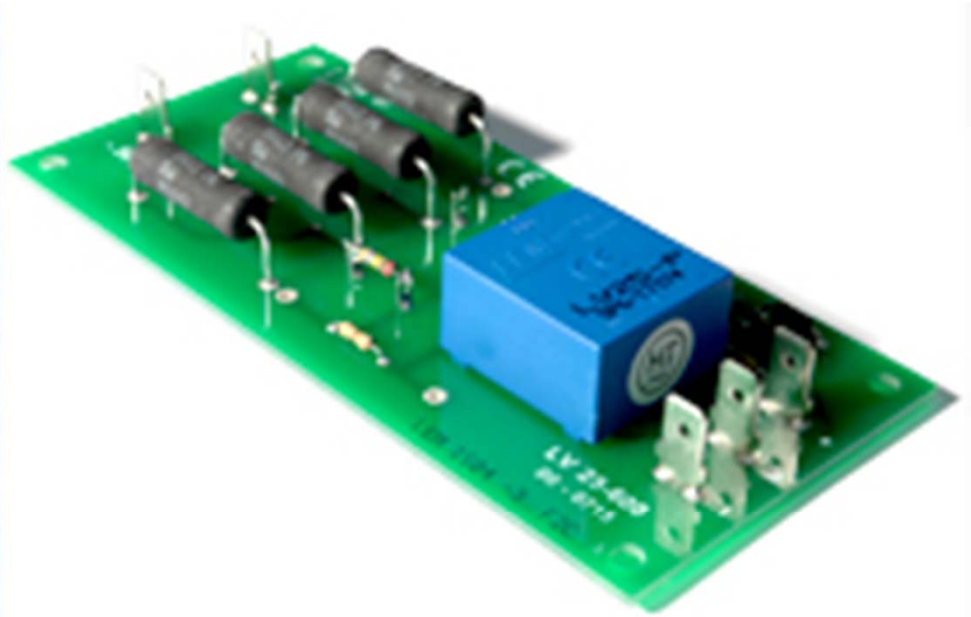
5Pcs

10 €

ACS 712

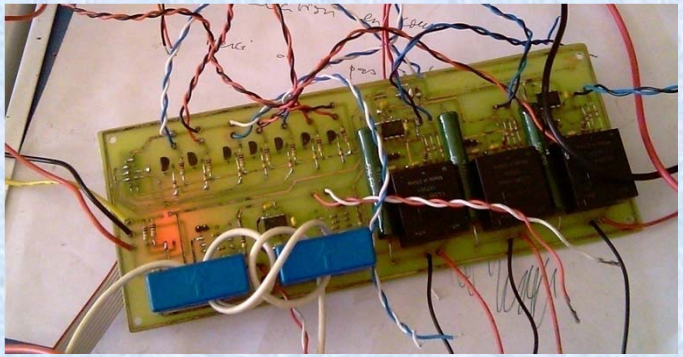
Voltage sensors

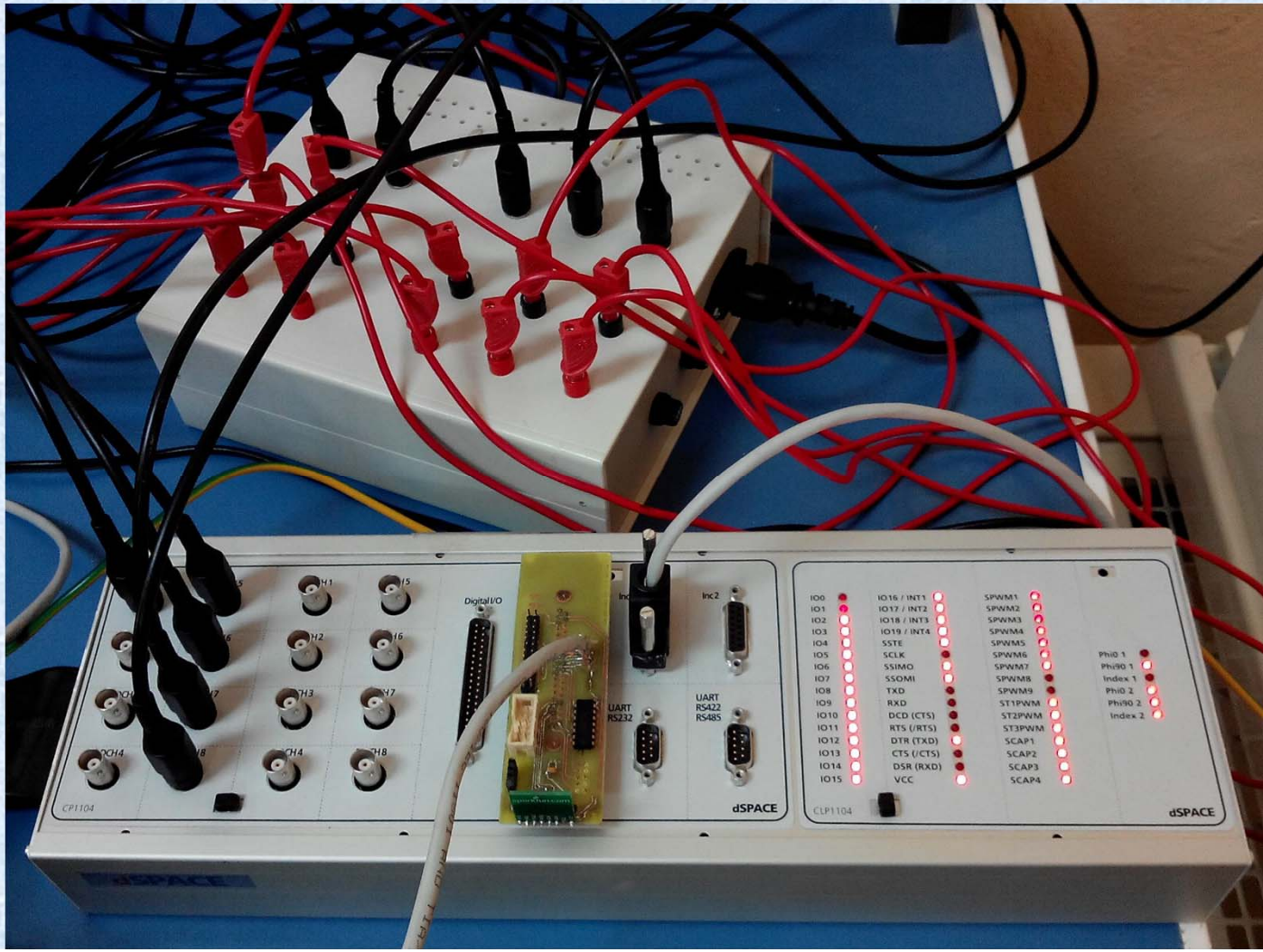
- Hall effect technology
- Galvanic isolation
- Measures in:
  - Continuous
  - Alternate
  - Instantaneous
- Needs external power resistances
- V → V
- Filtering
- ADC



Measure board

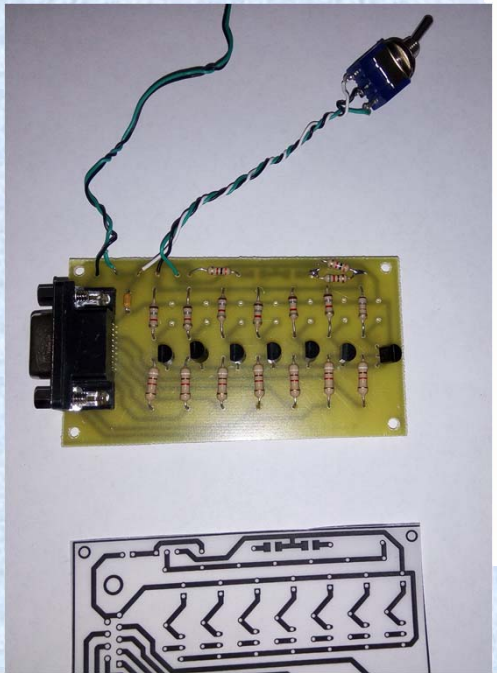
- Hall effect sensors
- Measure 2 courants
  - Iar, Ibr
- Measure 3 voltages
  - Vdc, Vas, Vbs
- Application: DFIG

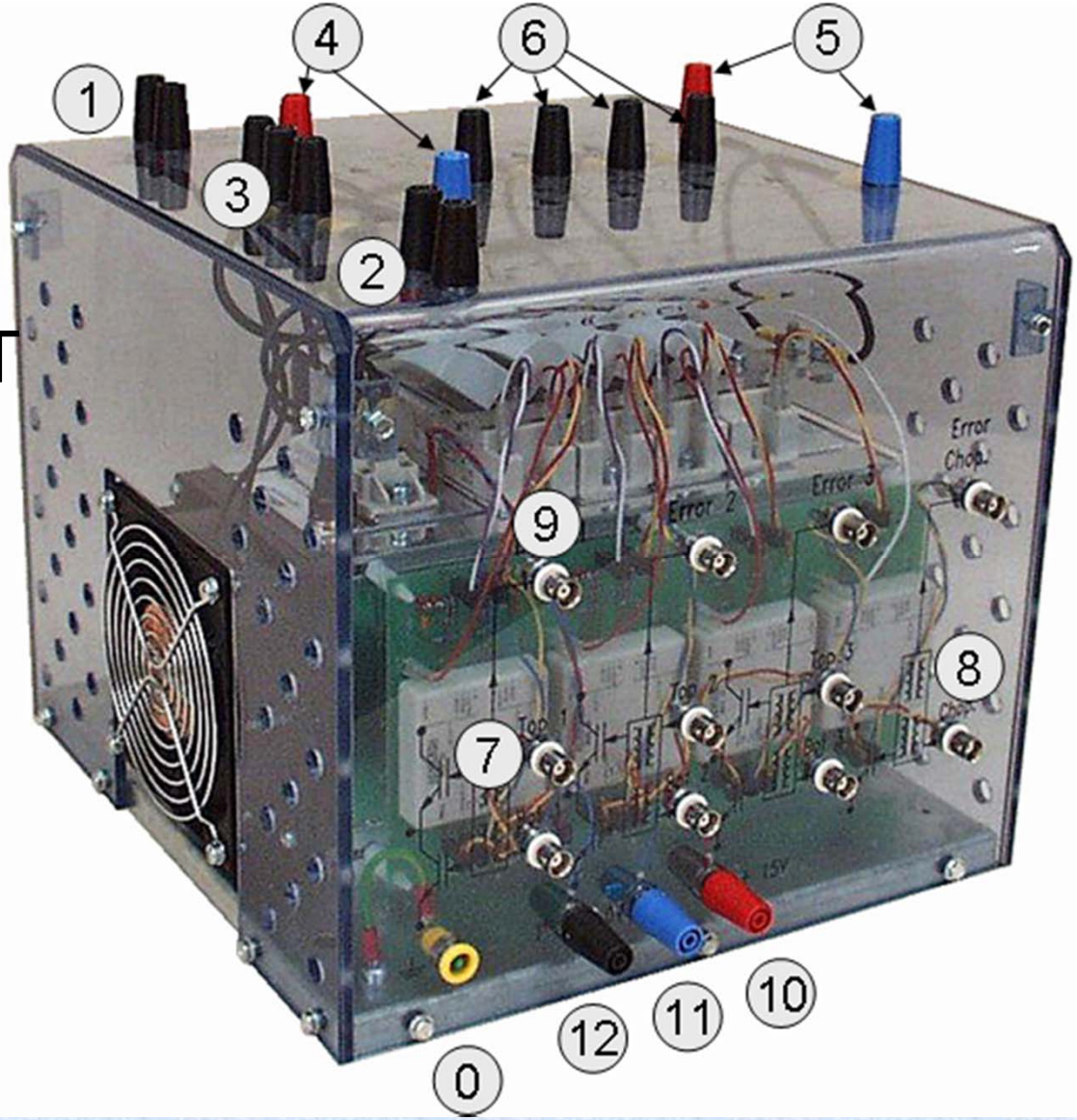




Voltage Fed Inverter

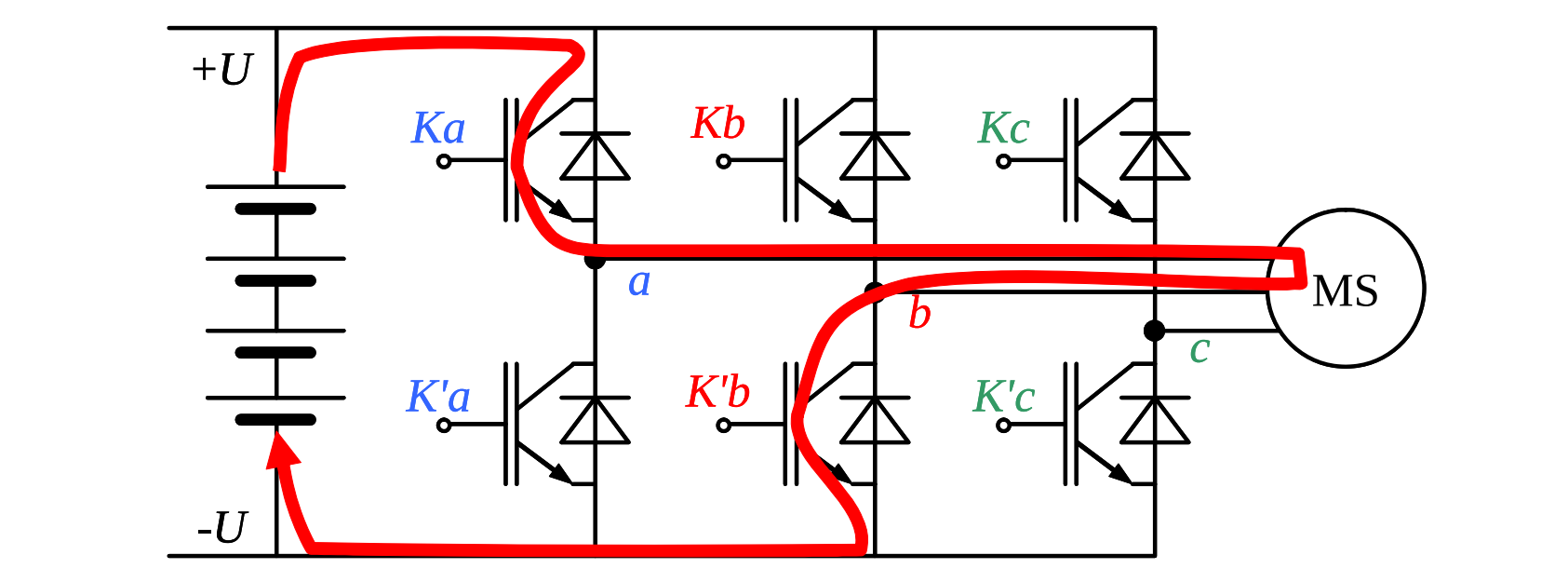
- Semikron 20 kVA
- Control of 3 legs: 6 IGBT
- PWM 0-15V drivers SKHI 22



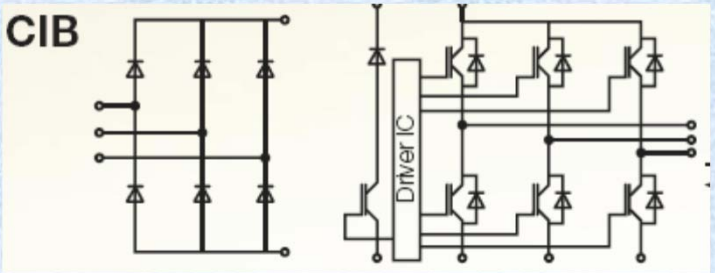


Voltage Fed Inverter

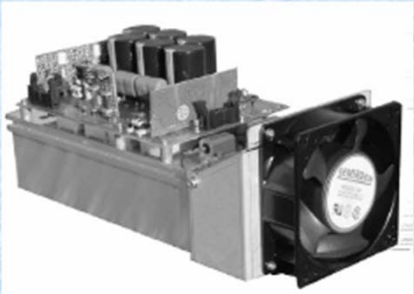
- IGBT or MOSFET inverter
- DC bus voltage
- BLDC, Vector or V/f control

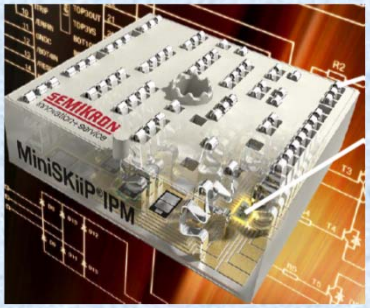


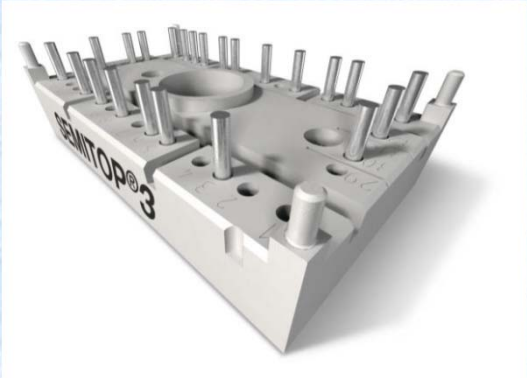
Power Electronics offers



- SEMIKRON
  - MiniSkiip IPM
  - SEMITOP 3 (IGBT 600V 12A – 174A)
  - SEMISTACK (IGBT)



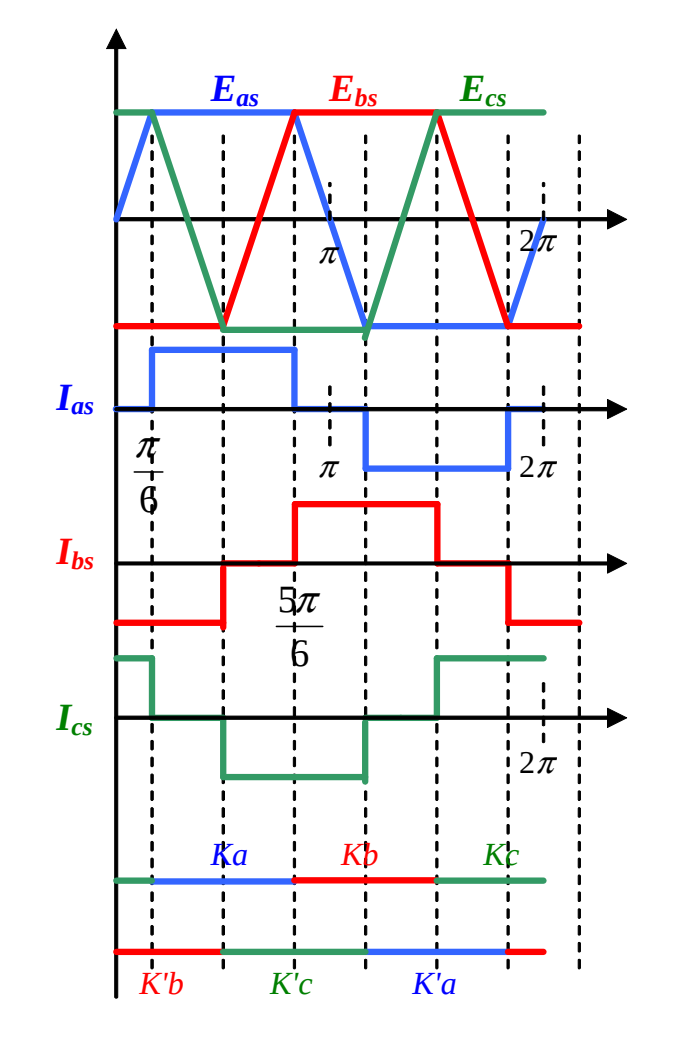




- IR : International Rectifier
  - 2131 drivers 3 high + 3 low integrated within a 28 PDIP ou SOIC, output 10-20V
  - IRFR2407 MOSFET V_{ds}=75V, I_d=42 A, $R_{ds\ on}$=26mΩ

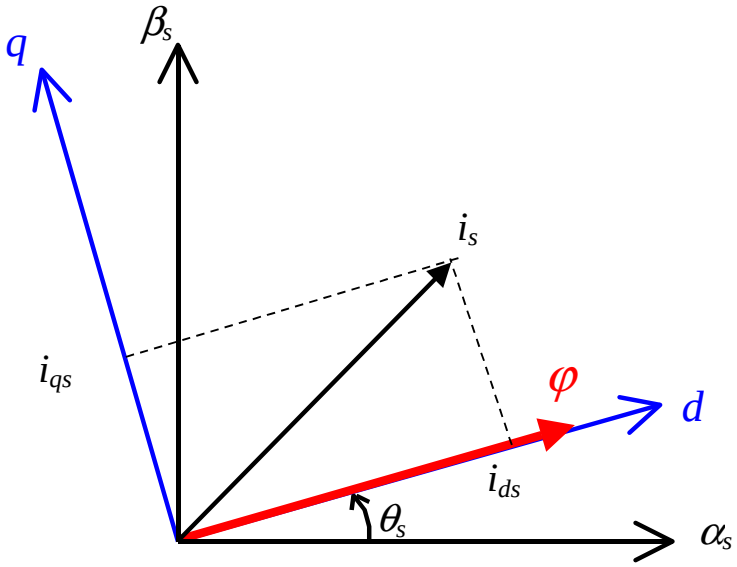
SM - BLDC

- Trapezoidal back e.m.f.
- 6 sectors control
- Rotor position known by 60°
- $C_e = k\Phi\, I$
- DC current control → BLDC



Vector control principle

- Transient model of the machine
- How to simplify the cross product torque formula
- Park reference frame orientation: Machine flux on the d-axis



(IM: Induction Motor)

$$C_e = \mathrm{p}\frac{M}{L_r}(\varphi_{dr} i_{qs} - \varphi_{qr} i_{ds})$$

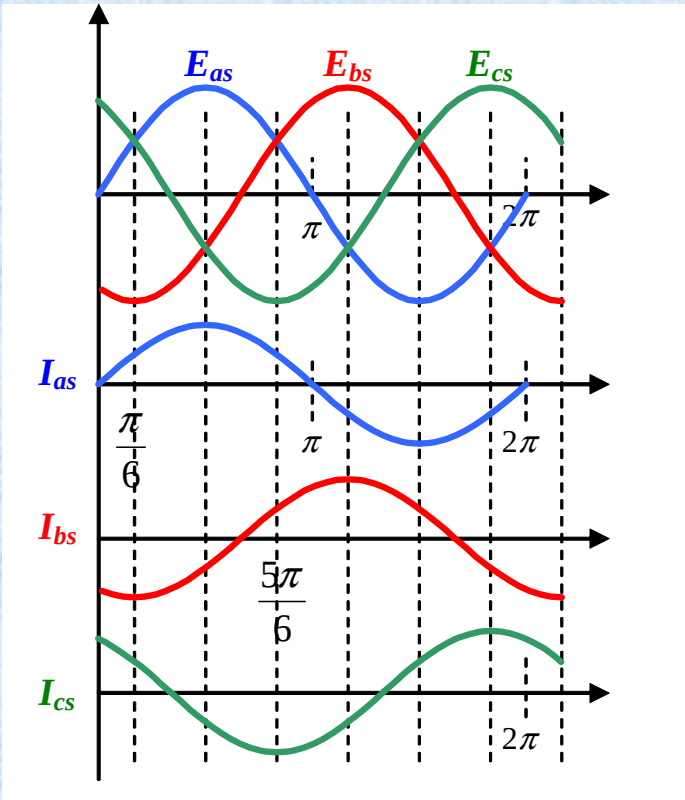
$$\Rightarrow \quad C_e = \mathrm{p}\frac{M}{L_r}(\varphi_{dr} i_{qs} - \cancel{\varphi_{qr} i_{ds}})$$

(SM: Synchronous Motor)

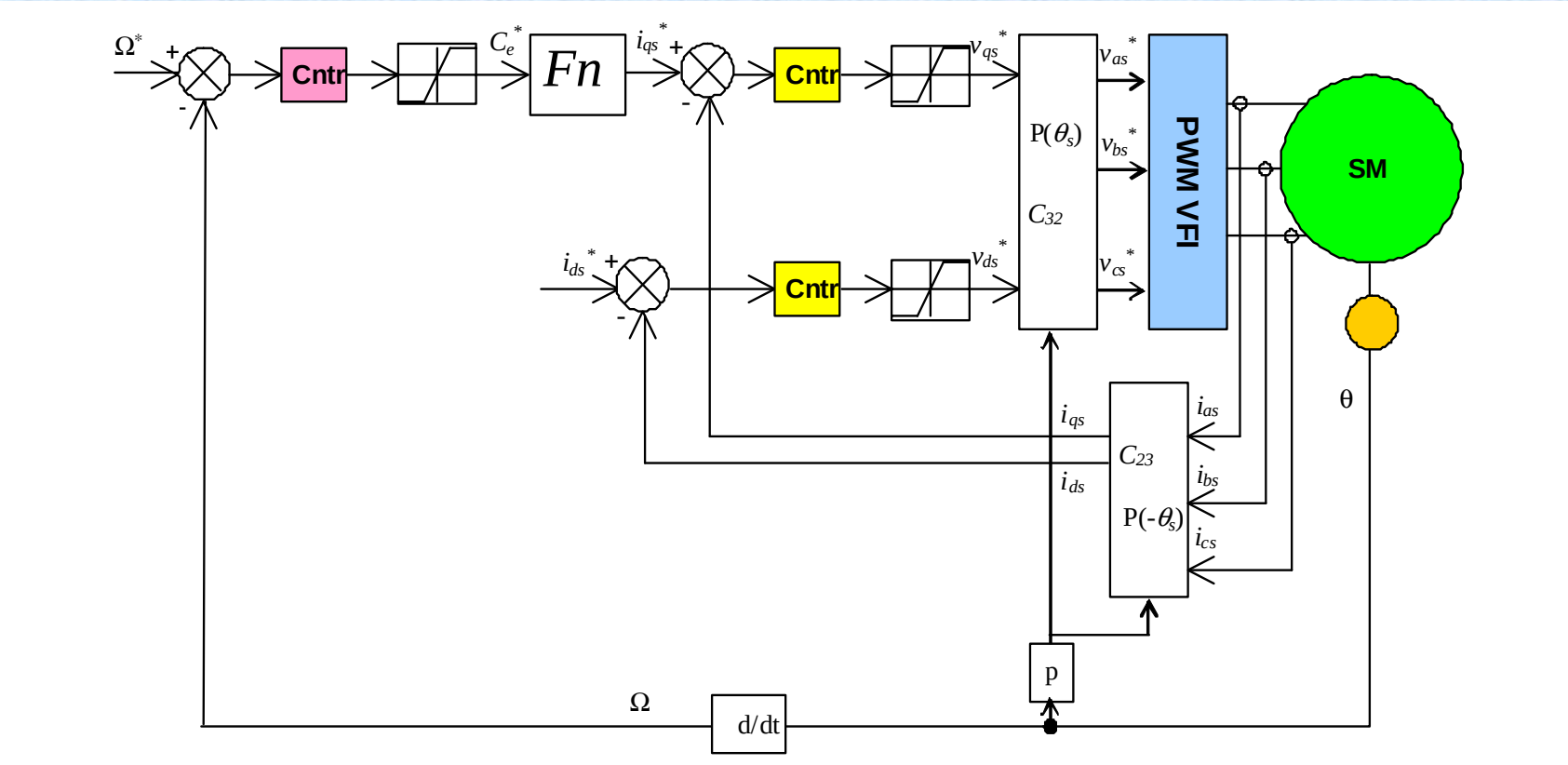
$$C_e = \mathrm{p}\left(\left(L_d - L_q\right) I_{ds} I_{qs} + \Psi_f I_{qs}\right)$$

Vector control of SM

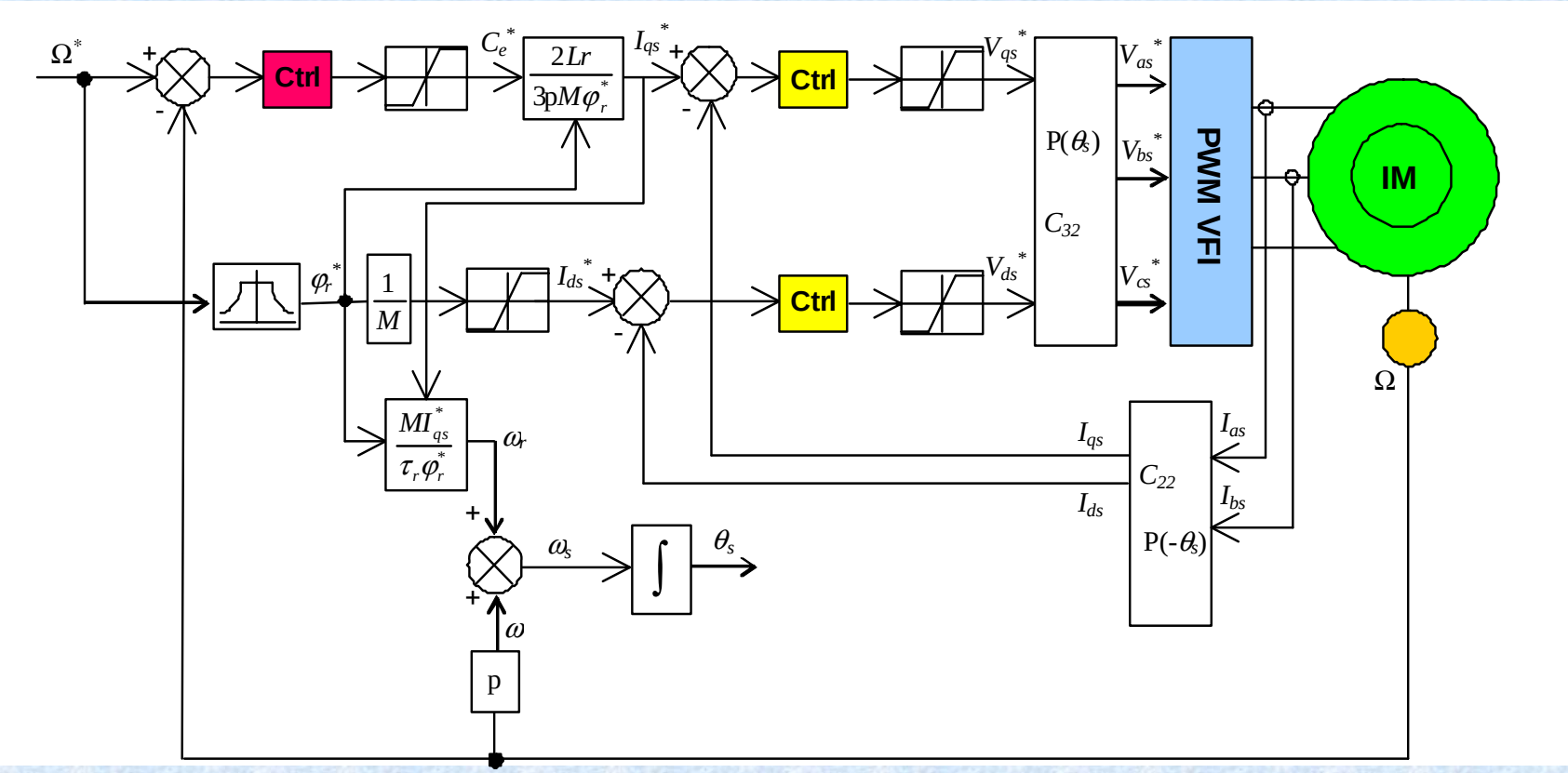
- Sinusoidal back e.m.f.
- Rotor position to be exactly know
- Needs computation: Park transformation
- $C_e = \mathrm{p}\left(\left(L_d - L_q\right) I_{ds} I_{qs} + \Psi_f I_{qs}\right)$
- PWM or hysteresis current control



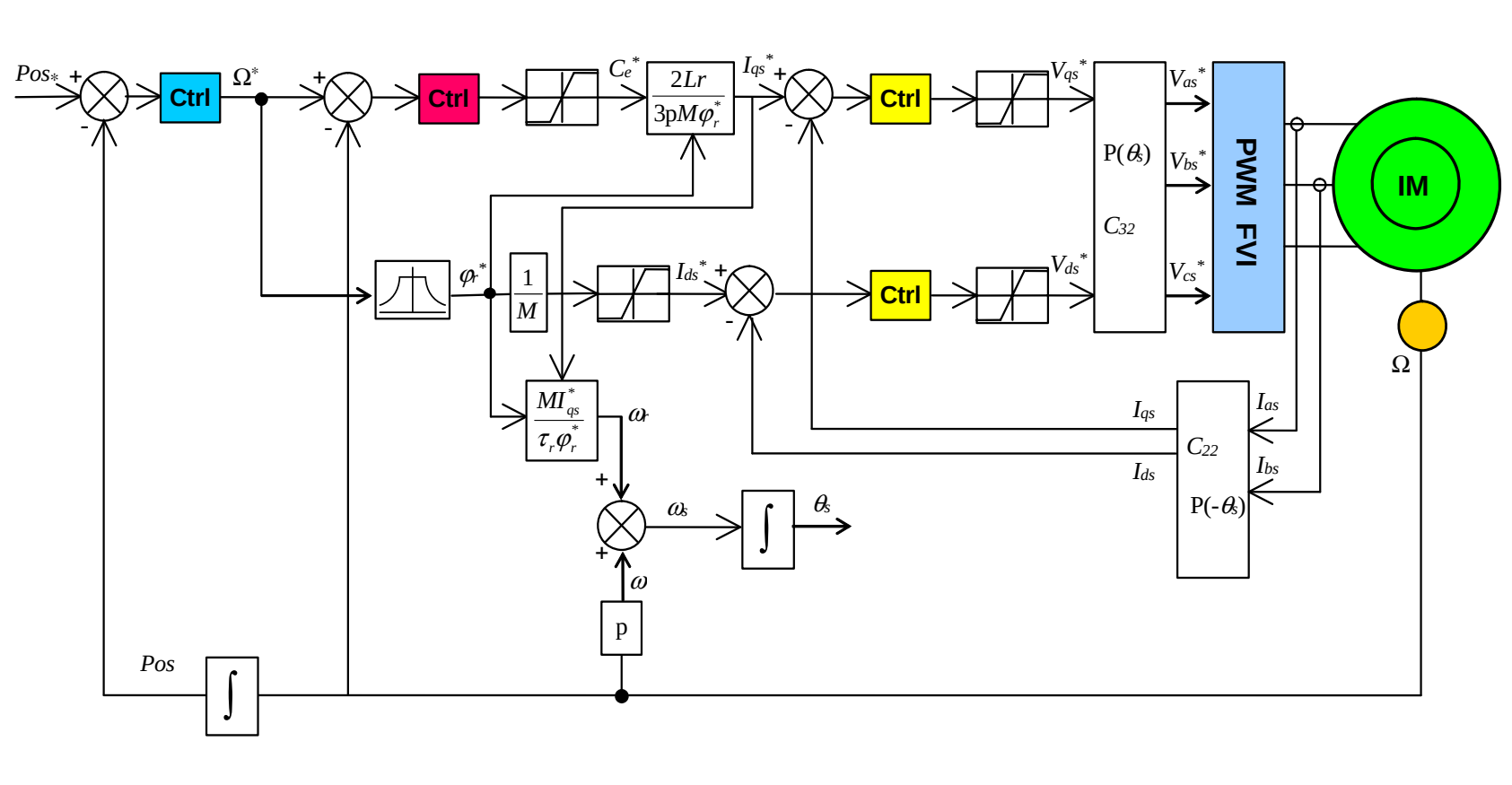
SM control scheme: Speed control



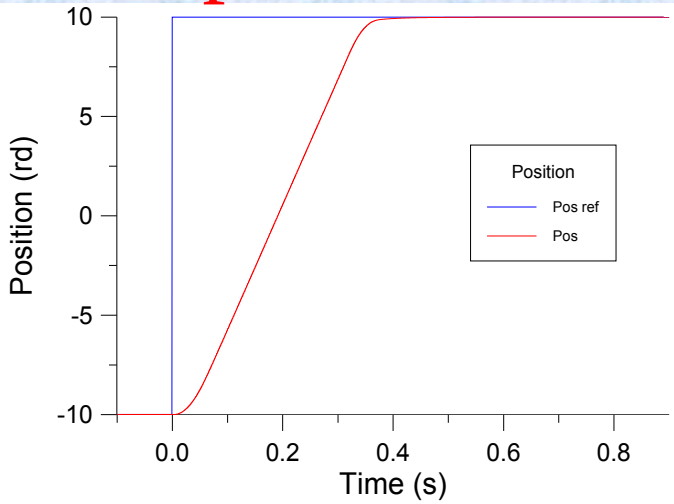
IM IRFO control scheme: Speed control

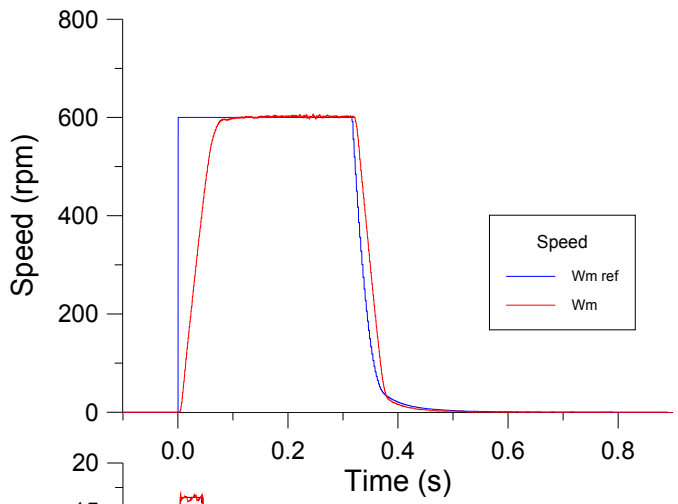


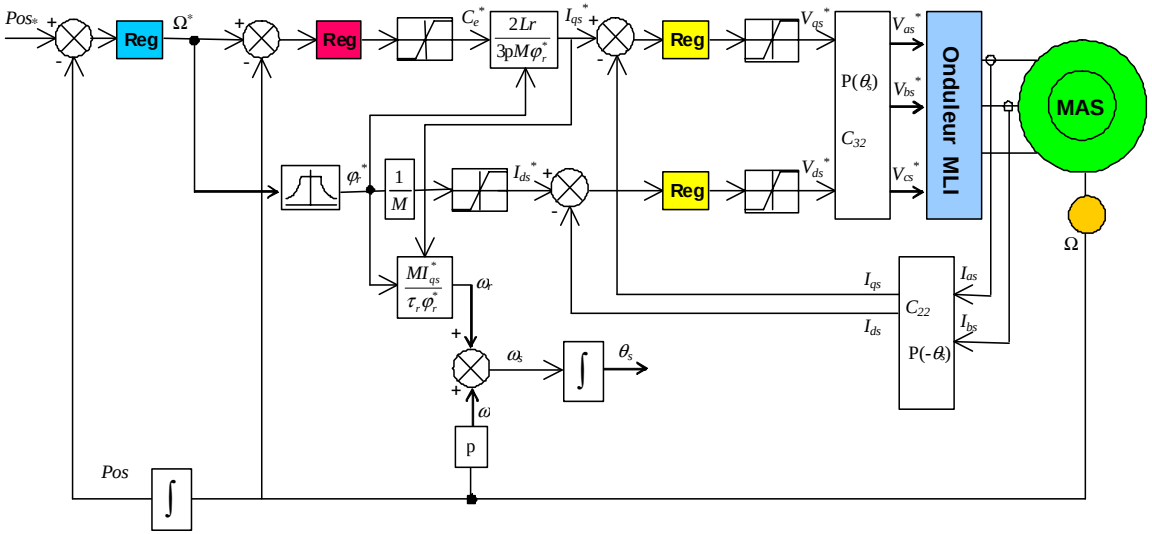
IM IRFO control scheme: Position control

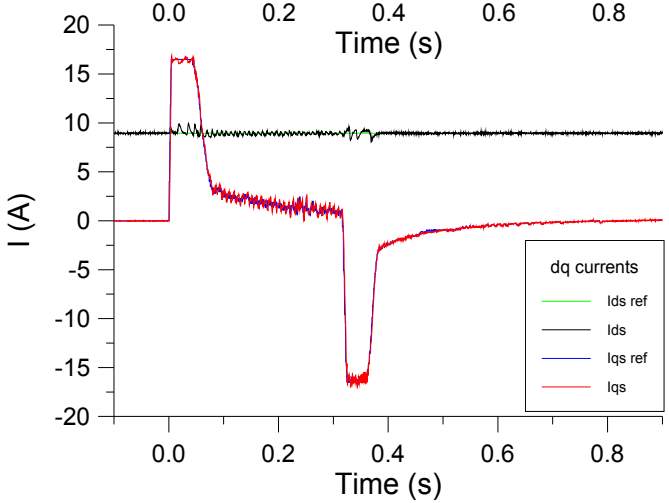


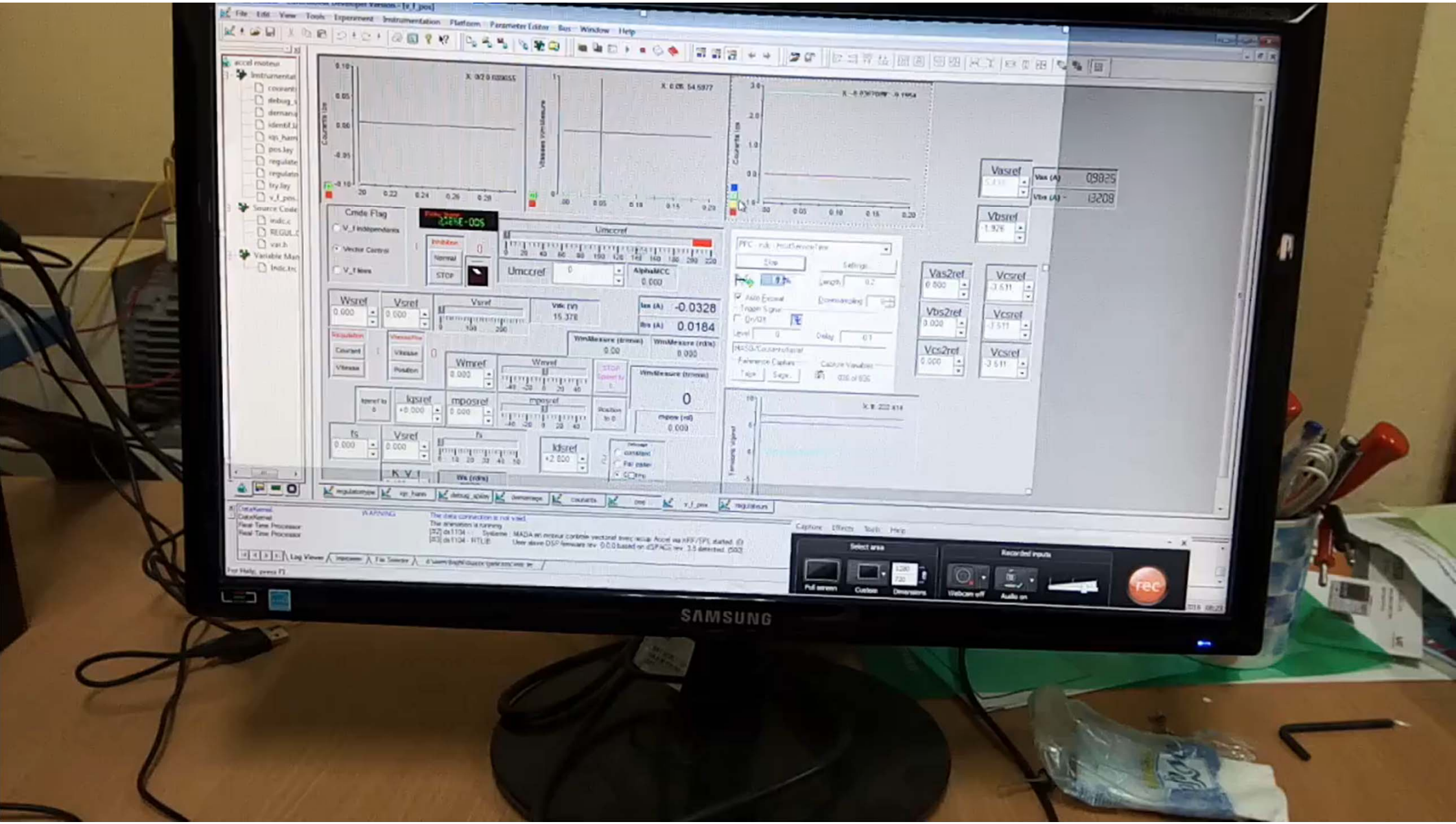
Capture, position control (IM)



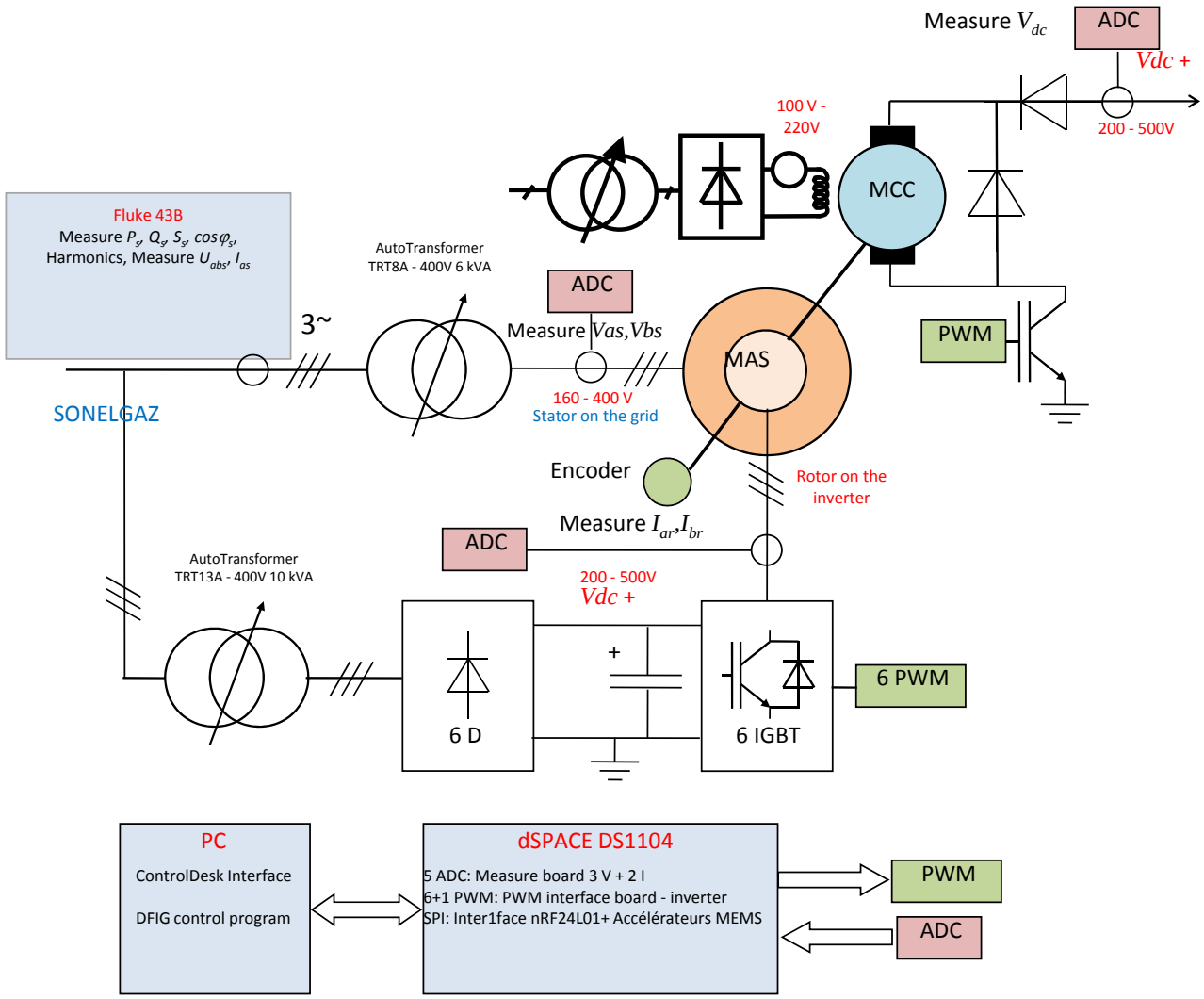


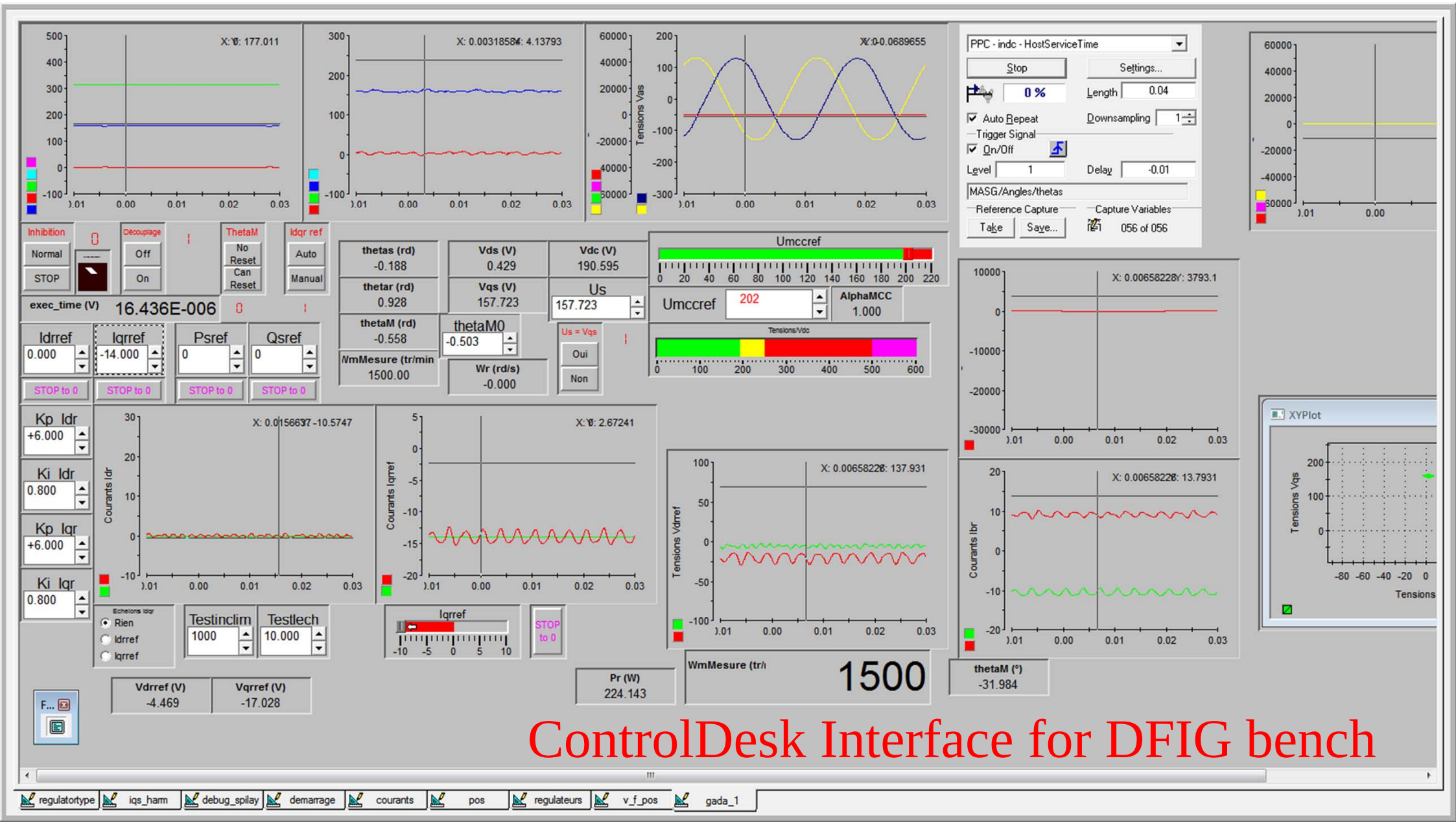





Cmde Flag
Vector Control
STOP
Umccref
Wsref
Vsref
Wmref
Wmref
kpsref
mposref
fs
Vsref
kdsref
Vasref
Vbsref
Vas2ref
Vcsref
Vbs2ref
Vcs2ref
Vcsref
Vcsref
Ias (A) -0.0328
Ibs (A) 0.0184
0
rec
SAMSUNG

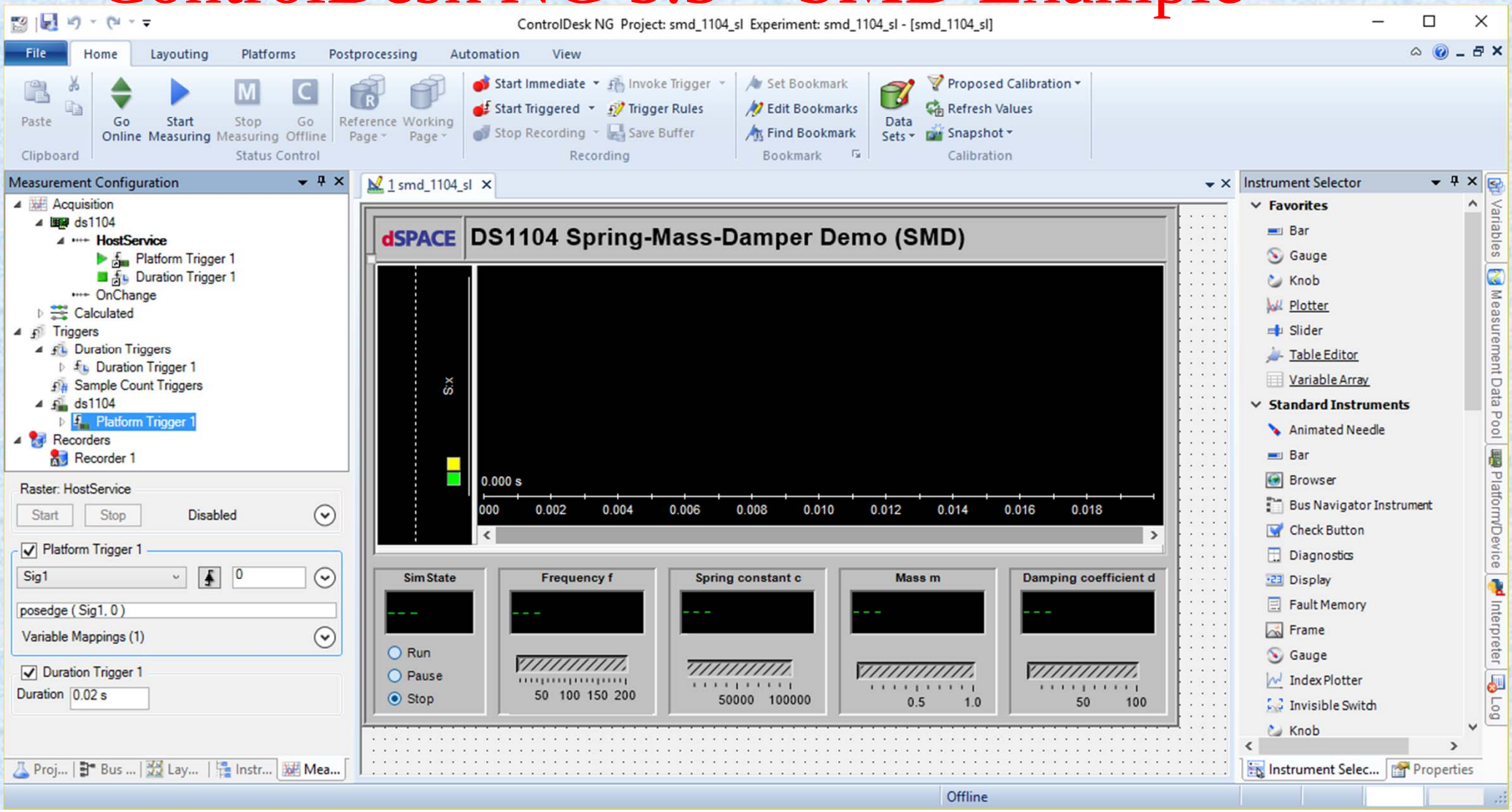
DFIG bench scheme





ControlDesk Interface for DFIG bench

ControlDesk NG 5.3 – SMD Example



dSPACE developement

- Matlab/Simulink
  - Draw the model (MDL)
  - Generate C code, Embedded Coder/ build
  - C compiling, linking
  - Loading, animation / ControlDesk
- Handcoding (langage C)
  - Write C source code
  - C compiling, linking
  - Loading, animation / ControlDesk

Tutorial

Abdallah Darkawi, Initiation au systeme dSPACE, de l'interface RTI dans Simulink et de ControlDesk Next Generation 5.4

version 2.0

Ecole d'ingenieur. France. 2016. <cel-01259564v2>

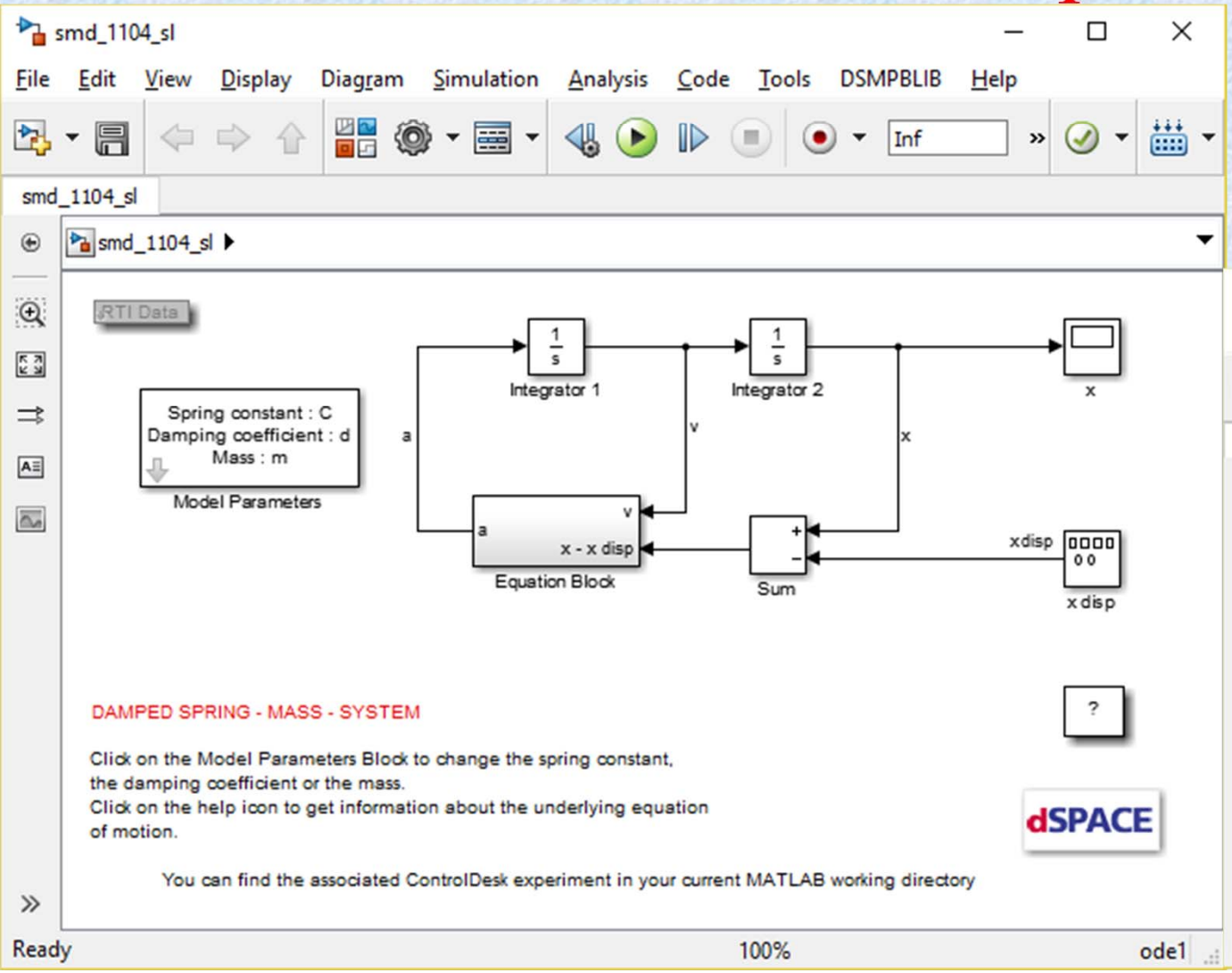
https://cel.archives-ouvertes.fr/cel-01259564v2/document

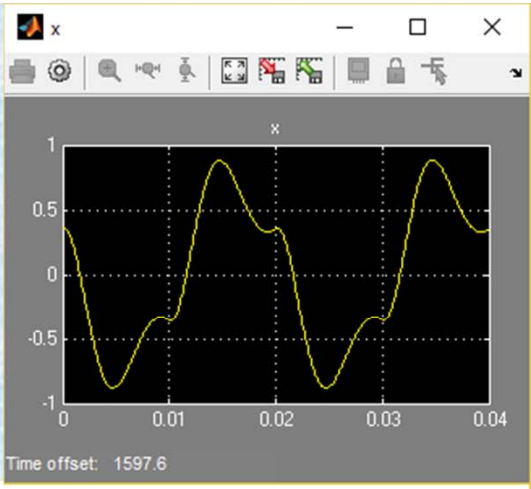
Matlab / Simulink configuration

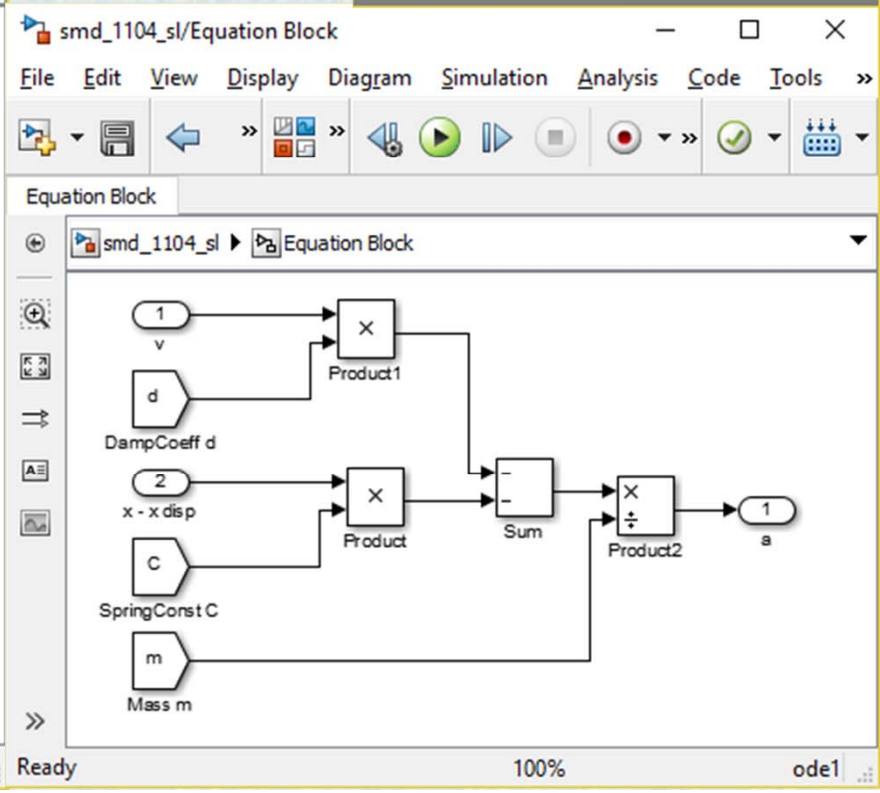
```
================================================================================
Configuring dSPACE(R) Software for MATLAB(R) 8.3.0.532 (R2014a) ...

RTI              Real-Time Interface to Simulink (RTI1104)  7.3         02-Nov-2014 okay
RTIFPGA          RTI FPGA Programming Blockset              2.8         02-Nov-2014 okay
DSMPBLIB         Model Port Block Library                   2.2         02-Nov-2014 okay
DSRT             dSPACE Real-Time Target (DSRT)             2.2         02-Nov-2014 okay
DSMSBLIB         Model Separation Block Library             2.1.1       02-Nov-2014 okay
DSMLCON24        dSPACE MATLAB Connection 2.4 (win64)       2.4         01-Jun-2012 okay
================================================================================

*** RTI Platform Support RTI1104 activated.
    Wizard: Click here to re-enable RTI platform selection for next MATLAB start.
```

Simulink – SMD Example

Without DS1104 RTI







Handcoded– SMD Example

Without DS1104 RTLib

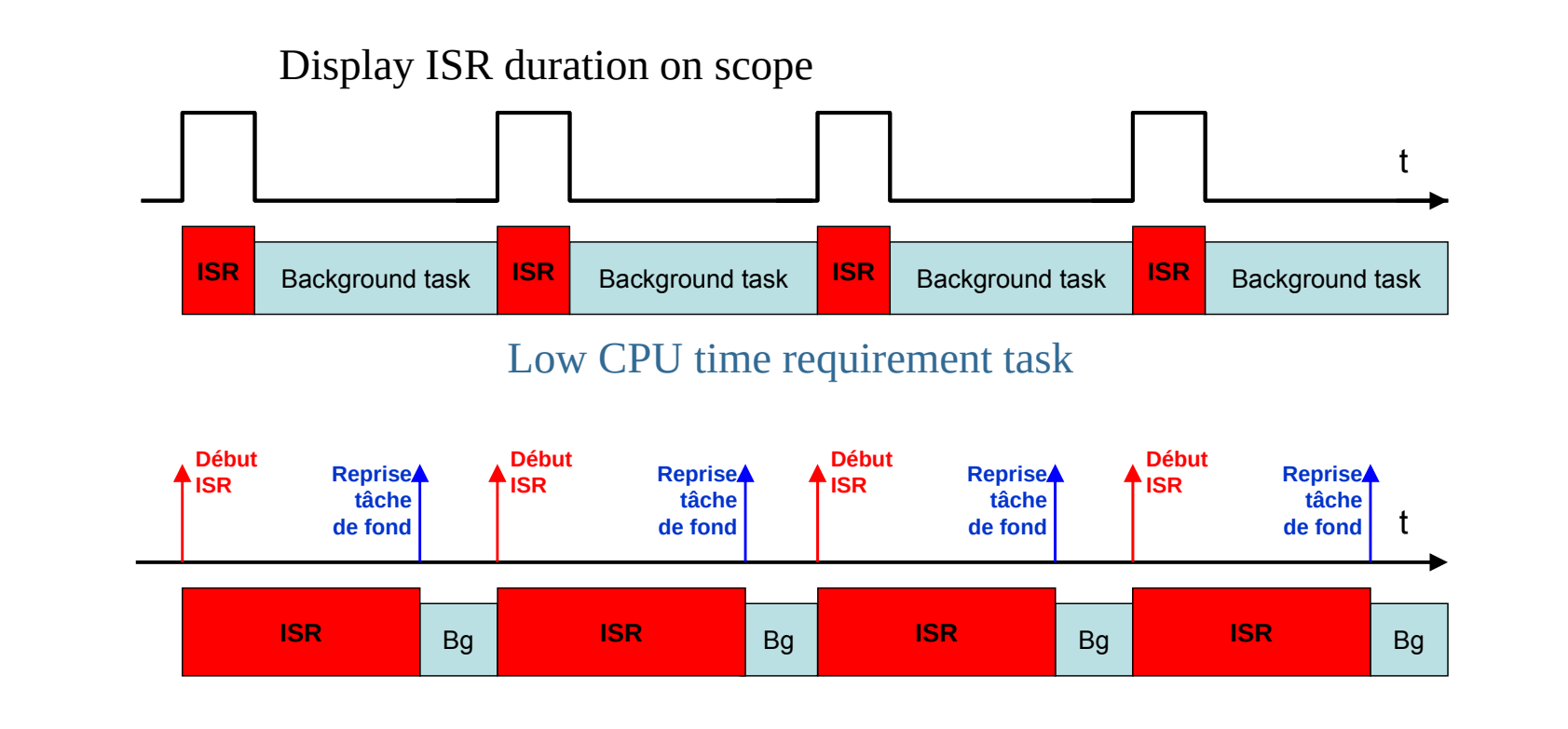
```
/*****************************************************************************
* FILE:
*   smd_1104_hc.c
* RELATED FILES:
*   Brtenv.h
* DESCRIPTION:
*  PPC application demo program for the DS1104 processor board
*  Simulation of a continuous 2nd-order spring-mass-damper system.
* dSPACE GmbH, Technologiepark 25, 33100 Paderborn, Germany
*
* $RCSfile: smd_1104_hc.c $ $Revision: 1.4 $ $Date: 2012/01/18 15:05:26GMT+01:00 $
*****************************************************************************/

#include <Brtenv.h>

/*---------------------------------------------------------------------------*/

#define input(x_disp) {                        \
  static Int32 i = 0;                          \
  static Int32 step_count = 0;                 \
                                               \
  step_count = (Int32)( 1/(2*f*DT) );          \
  if (++i >= step_count)                       \
  {                                            \
    x_disp *= -1;                              \
    i = 0;                                     \
  } }

/* empty output macro */
#define output(y)

/*---------------------------------------------------------------------------*/

#define DT 1.0e-4                            /* 100 us simulation step size */

#define NSTATES 2                                      /* number of states */
```

```c
#define DT 1.0e-4                           /* 100 us simulation step size */

#define NSTATES 2                                     /* number of states */
Float64 x_[NSTATES], x_dot[NSTATES];                    /* state matrices */

#define D_DEF 16.0                                   /* initial damping */
#define C_DEF 20000.0                         /* initial spring coefficient */
#define M_DEF 0.04                                     /* initial mass */
#define F_DEF 50                                   /* initial frequency */

volatile Float64 d;                                          /* damping */
volatile Float64 c;                                  /* spring coefficient */
volatile Float64 m;                                             /* mass */
volatile Float64 f;                                        /* frequency */

Float64 x_disp = 0.5;                       /* excitation, amplitude = 0.5 */
Float64 x;                                                   /* position */
Float64 v;                                                   /* velocity */
Float64 a;                                                /* acceleration */

/* variable for execution time profiling */
Float64 exec_time;                                      /* execution time */

/*-----------------------------------------------------------------------*/

void derivatives(void)
{
  Float64 fc, fd;

  fd       = (d) * x_[1];                             /* damping function */
  fc       = (c) * (x_[0] - x_disp);                   /* spring function */
  x_dot[0] = x_[1];                                  /* first derivative */
  x_dot[1] = (-fd - fc) / (m);                      /* second derivative */
}

/*-----------------------------------------------------------------------*/

void integrate_with_euler(void)
{
  Int32 i;
```

```c
void integrate_with_euler(void)
{
  Int32 i;

  derivatives();
  for (i = 0; i < NSTATES; i++)
    x_[i] = x_[i] + x_dot[i] * DT;
}

/*-----------------------------------------------------------------------*/

void copy_results(void)
{
  x = x_[0];
  v = x_dot[0];
  a = x_dot[1];
}

/*-----------------------------------------------------------------------*/

void isr_srt(void)
{
  ts_timestamp_type ts;
  RTLIB_SRT_ISR_BEGIN();                                /* overload check */
  ts_timestamp_read(&ts);
  host_service(1,&ts);                          /* Data Acquisition service */

  RTLIB_TIC_START();                              /* start time measurement */

  input(x_disp);                                        /* get input value */
  output(x_[0]);                            /* output result (x[0] is output) */
  integrate_with_euler();                          /* update state matrices */
  copy_results();

  exec_time = RTLIB_TIC_READ();
  RTLIB_SRT_ISR_END();                                  /* overload check */

}

/*-----------------------------------------------------------------------*/
```

ISR Programing

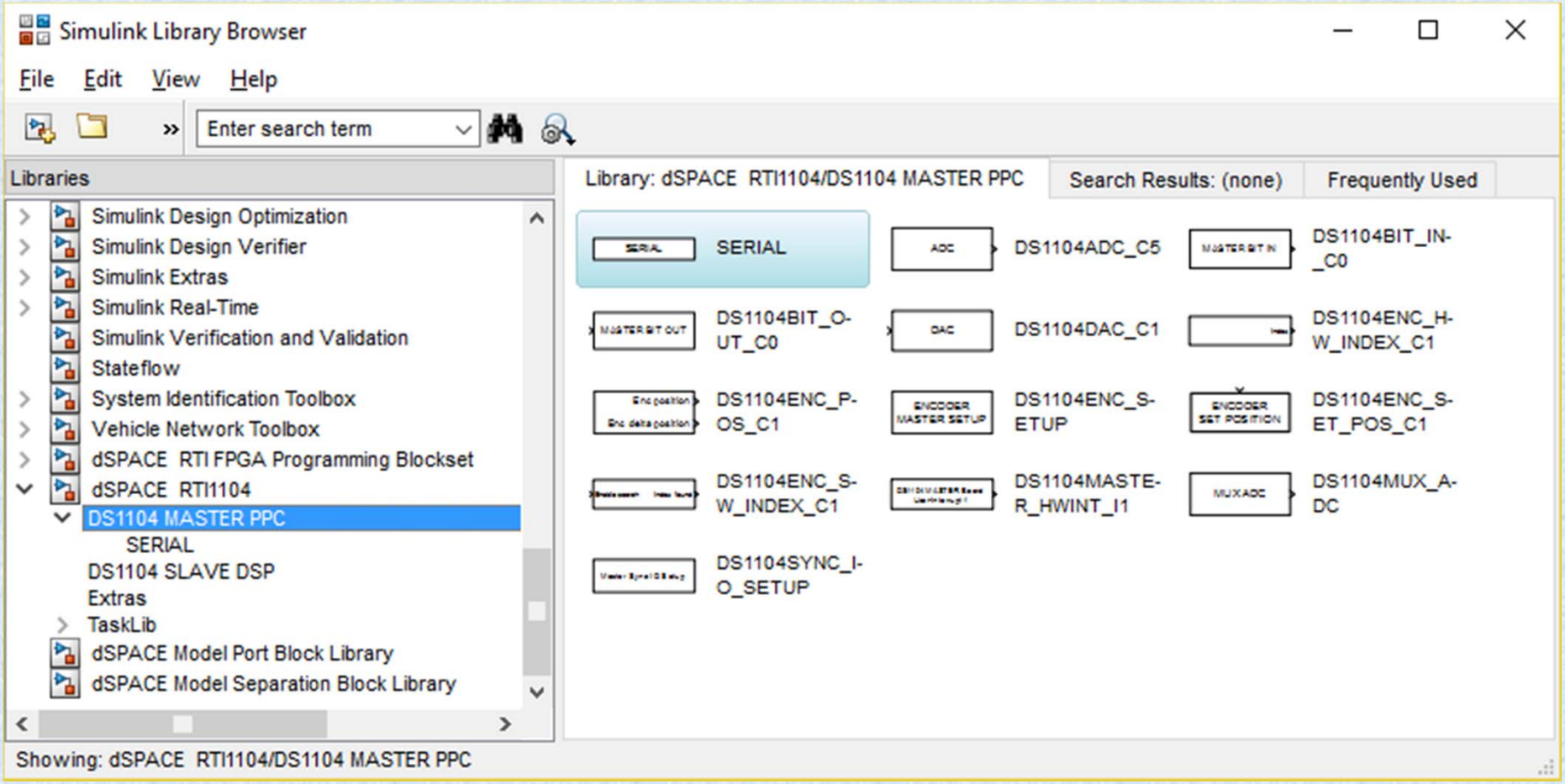
ISR Interrupt Service Routine : PWM_sync_interrupt()



```c
void main(void)
{

  Int32 i;


  d = D_DEF;                                        /* initialize parameters */
  c = C_DEF;
  m = M_DEF;
  f = F_DEF;

  for (i = 0; i < NSTATES; i++)                     /* clear state variables */
    x_[i] = 0.0;

  init();                              /* DS1104 and RTLib1104 initialization */

  msg_info_set(MSG_SM_RTLIB, 0, "System started.");

  RTLIB_SRT_START(DT, isr_srt);                   /* start sample rate timer */

  while(1)
  {

    RTLIB_BACKGROUND_SERVICE();                       /* background service */

  }

}
```

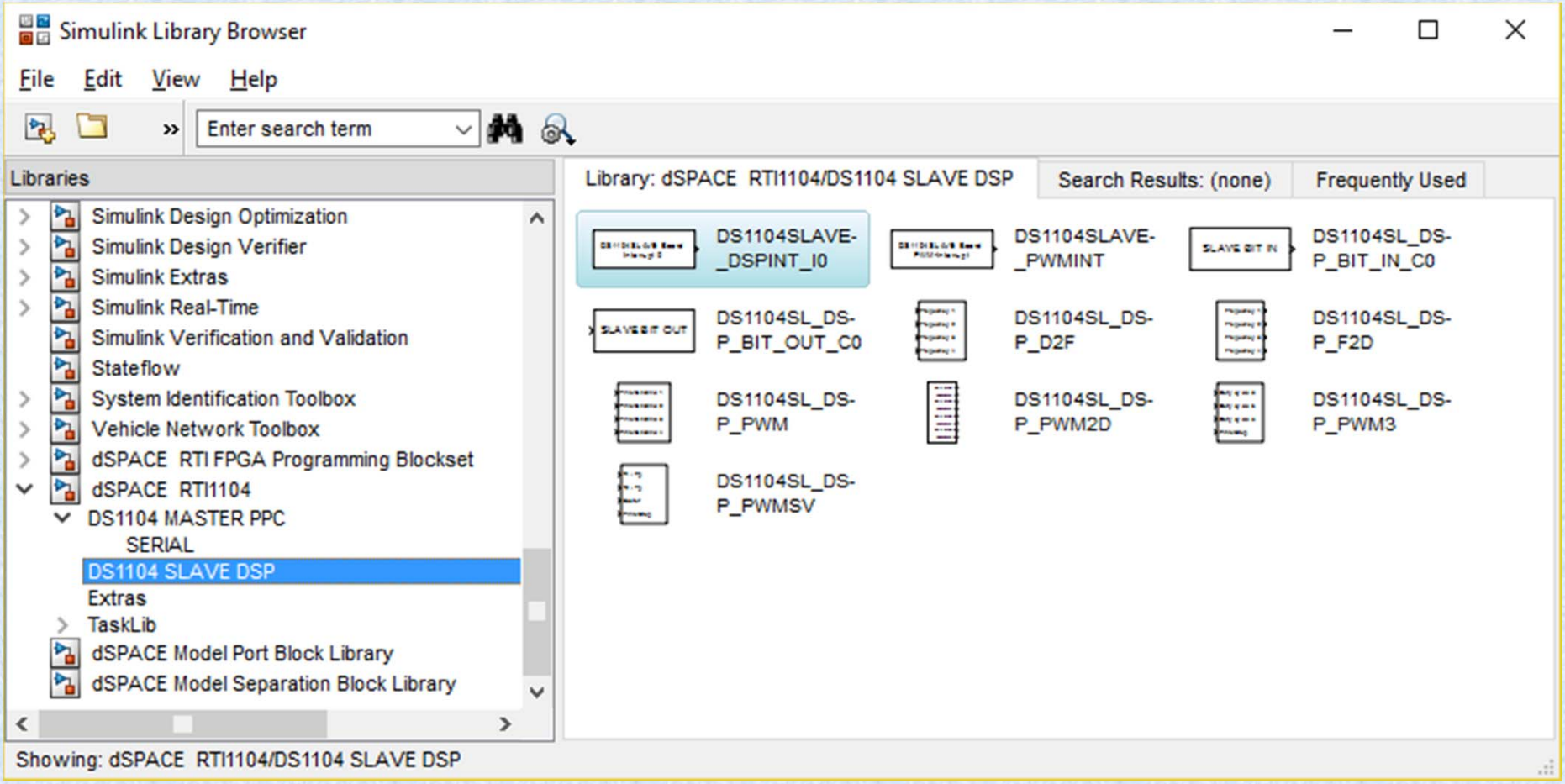
Tools Embedded Coder de Matlab / Simulink

DS1104 RTI Master

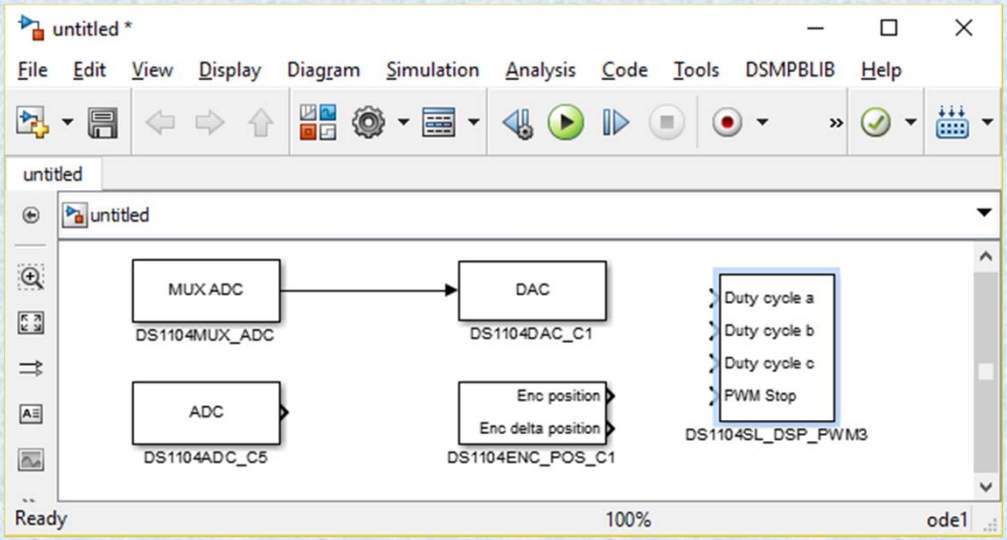


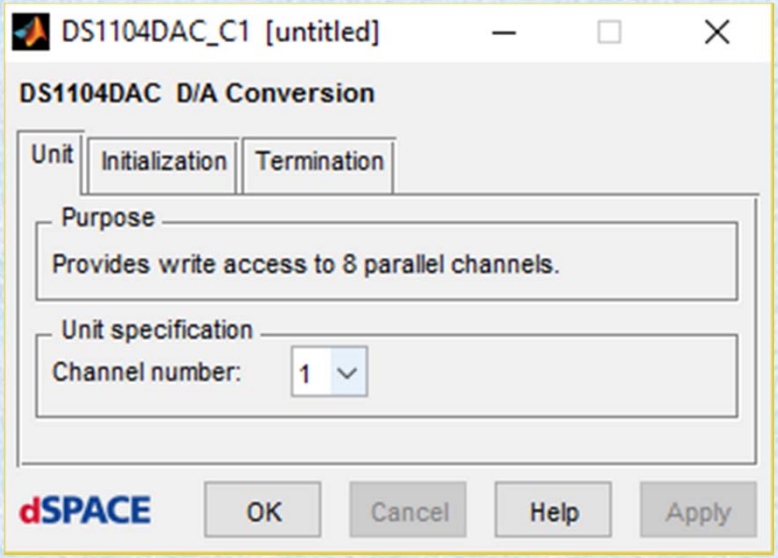
Tools Embedded Coder de Matlab / Simulink

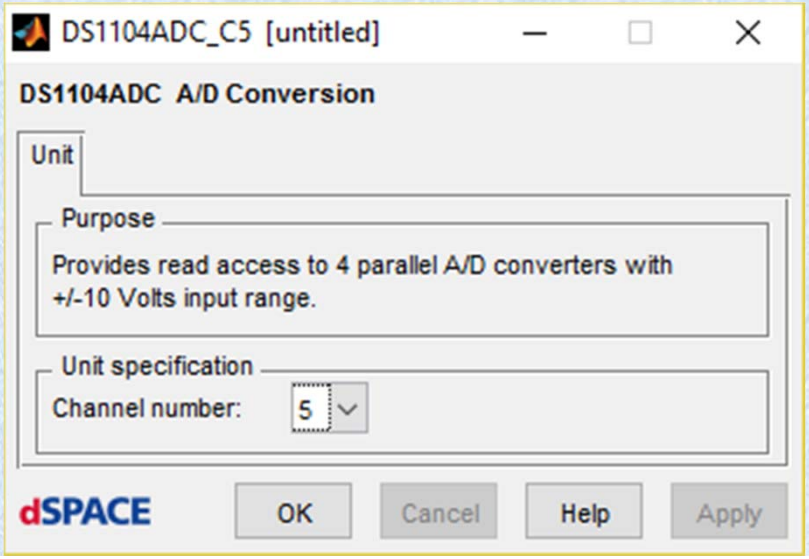
DS1104 RTI Slave DSP F240

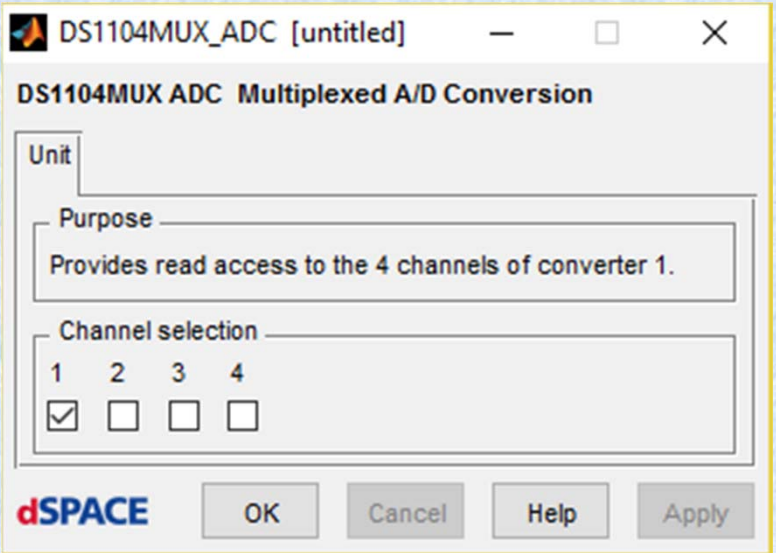


DS1104 RTI

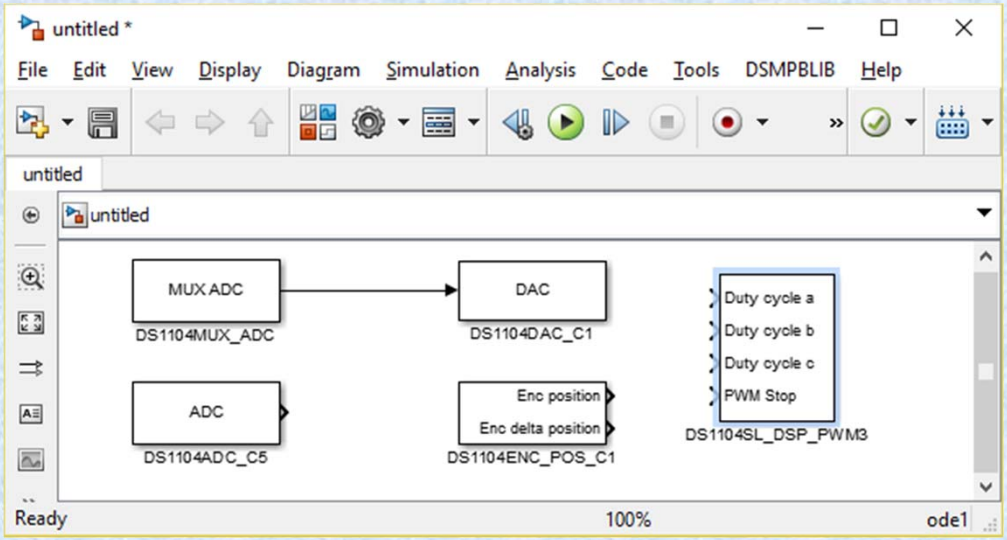

untitled *
File Edit View Display Diagram Simulation Analysis Code Tools DSMPBLIB Help
untitled
untitled
MUX ADC
DS1104MUX_ADC
DAC
DS1104DAC_C1
Duty cycle a
Duty cycle b
Duty cycle c
PWM Stop
DS1104SL_DSP_PWM3
ADC
DS1104ADC_C5
Enc position
Enc delta position
DS1104ENC_POS_C1
Ready
100%
ode1

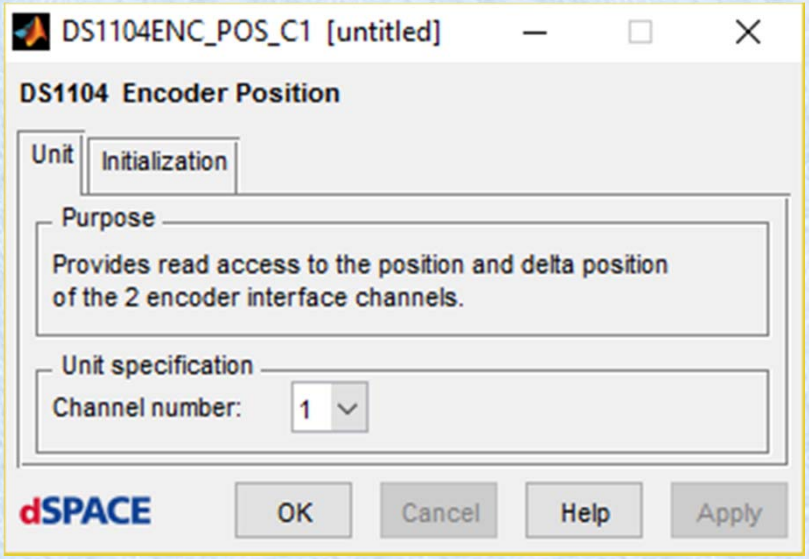

DS1104DAC_C1 [untitled]
DS1104DAC D/A Conversion
Unit
Initialization
Termination
Purpose
Provides write access to 8 parallel channels.
Unit specification
Channel number:
1
dSPACE
OK
Cancel
Help
Apply

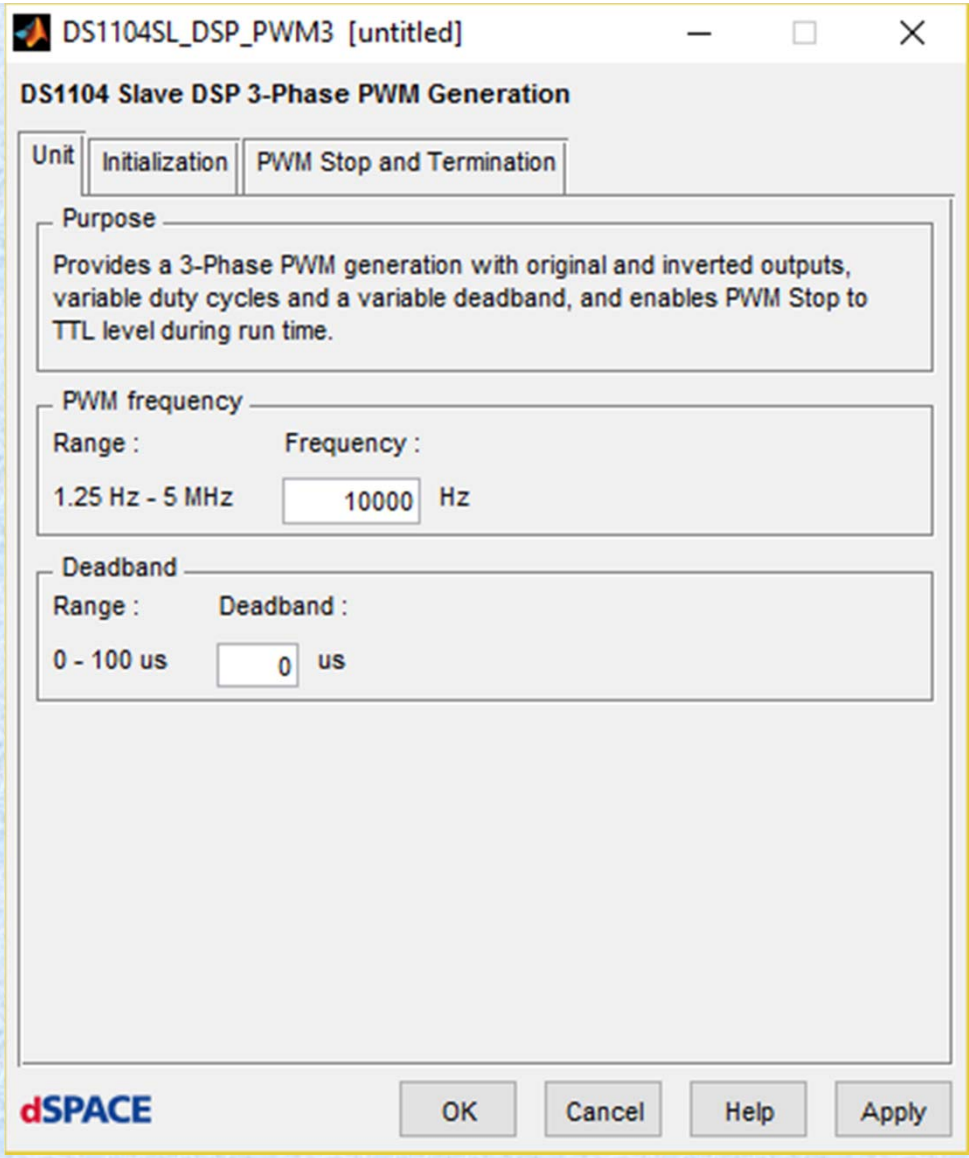

DS1104ADC_C5 [untitled]
DS1104ADC A/D Conversion
Unit
Purpose
Provides read access to 4 parallel A/D converters with +/-10 Volts input range.
Unit specification
Channel number:
5
dSPACE
OK
Cancel
Help
Apply


DS1104MUX_ADC [untitled]
DS1104MUX ADC Multiplexed A/D Conversion
Unit
Purpose
Provides read access to the 4 channels of converter 1.
Channel selection
1 2 3 4
dSPACE
OK
Cancel
Help
Apply

DS1104 RTI


untitled *
File Edit View Display Diagram Simulation Analysis Code Tools DSMPBLIB Help
untitled
untitled
MUX ADC
DS1104MUX_ADC
DAC
DS1104DAC_C1
Duty cycle a
Duty cycle b
Duty cycle c
PWM Stop
DS1104SL_DSP_PWM3
ADC
DS1104ADC_C5
Enc position
Enc delta position
DS1104ENC_POS_C1
Ready
100%
ode1


DS1104ENC_POS_C1 [untitled]
DS1104 Encoder Position
Unit
Initialization
Purpose
Provides read access to the position and delta position of the 2 encoder interface channels.
Unit specification
Channel number:
1
dSPACE
OK
Cancel
Help
Apply


DS1104SL_DSP_PWM3 [untitled]
DS1104 Slave DSP 3-Phase PWM Generation
Unit
Initialization
PWM Stop and Termination
Purpose
Provides a 3-Phase PWM generation with original and inverted outputs, variable duty cycles and a variable deadband, and enables PWM Stop to TTL level during run time.
PWM frequency
Range :
Frequency :
1.25 Hz - 5 MHz
10000
Hz
Deadband
Range :
Deadband :
0 - 100 us
0
us
dSPACE
OK
Cancel
Help
Apply

DS1104 RTI

DS1104RTIReference.pdf

I/O characteristics

Scaling between the analog input voltage and the output of the block:

Input Voltage Range	Simulink Output
–10 V ... +10 V	–1 ... +1 (double)

Dialog pages

The dialog settings can be specified on the following pages:

- *Unit Page (DS1104MUX_ADC)* on page 18

Related RTLib functions

ds1104_adc_start, ds1104_adc_mux, ds1104_adc_read_mux, ds1104_adc_read_conv, ds1104_adc_trigger_setup

Unit Page (DS1104MUX_ADC)

Purpose

To specify the channels to be multiplexed.

Dialog settings

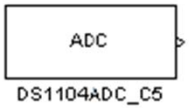
Channel selection Lets you select a set of up to 4 channels.

Related topics

References

- *DS1104MUX_ADC* on page 17

DS1104ADC_Cx



Purpose

To read from a single channel of one of 4 parallel A/D converter channels.

I/O mapping

For information on the I/O mapping, refer to *ADC Unit* (📖 *DS1104 Features*).

DS1104 RTLib

DS1104RTLibReference.pdf

ds1104_adc_read_ch_immediately

Syntax

`Float64 ds1104_adc_read_ch_immediately(UInt16 channel)`

Include file

`io1104.h`

Purpose

To read one A/D channel without polling the end-of-conversion flag.

Description

This function can be used in an ADC end-of-conversion interrupt service routine.

> Before using this function, the converter must be started by means of `ds1104_adc_start`. The multiplexed converter (channels 1 ... 4) must be set with `ds1104_adc_mux` and started by using `ds1104_adc_delayed_start`.

I/O mapping

For information on the I/O mapping, refer to *ADC Unit* (📖 *DS1104 Features*).

Parameters

channel Specifies the channel number within the range 1 ... 8.

Return value

This function returns the scanned value scaled as follows:

Input Voltage Range	Return Value Range
–10 V ... +10 V	–1.0 ... +1.0

Related topics

References

- *ADC Unit* (📖 *DS1104 Features*)
- *ds1104_adc_read_ch* on page 200
- *Elementary Data Types* on page 21
- *Interrupt Handling* on page 84

DS1104 RTLib

- DS1104 hardware access, RTLib
  - PWM : `ds1104_slave_dsp_pwm3_xxxx`
  - ADC : `ds1104_adc_xxxx`
  - I/O : `ds1104_bit_io_xxxx`
  - Codeur : `ds1104_inc_xxxx`
  - ISR : `ds1104_enable_hardware_int`
- Software, RTLib
  - ISR : `ds1104_set_interrupt_vector`

IM IRFO handcoded C

- Routines
  - Indc.C : main(), PWM_sync_interrupt
  - Regul.C : Park, ParkInv, (sin/cos), C32, C22t
    Calc_thetas, BoucleId, BoucleIq,
    BoucleVitesse, BouclePosition,
    RegulI_Control, Regul_Vitesse
  - Var.H : constants, parameters, variables

IM IRFO Indc.C : void main(void)

```
void main(void)
{
init();                                /* DS1104 and RTLib1104 initialization */
//il faut lancer les service 1 fois avant d'enclencher le controledesk et les interruptions (non dit ds la doc)
host_service(1, 0);
msg_info_set(0, 0, "Systeme : IRFO : controle vectoriel, started.");
// configure I/O
ds1104_bit_io_init( DS1104_DIO0_OUT | DS1104_DIO1_OUT);
ds1104_bit_io_clear(DS1104_DIO0 | DS1104_DIO1);
/* init incremental encoder channel 1 (CodeurCh) : input signal for channel 2 via RS422 */
ds1104_inc_init( CodeurCh, DS1104_INC_MODE_RS422);
ds1104_inc_set_idxmode( CodeurCh, DS1104_INC_IDXMODE_OFF);// no reset
// ds1104_inc_set_idxmode( CodeurCh, DS1104_INC_IDXMODE_ON); /* set reset on index for channel 1 and latch to 0
// int Index Ch1
ds1104_set_interrupt_vector(DS1104_INT_INC_ENC_CH1, (DS1104_Int_Handler_Type)
         &Encoder_int_handler, SAVE_REGS_ON);
ds1104_enable_hardware_int(DS1104_INT_INC_ENC_CH1);

/* config DAC */
ds1104_dac_init( DS1104_DACMODE_TRANSPARENT);

/* initialization of slave DSP communication */
ds1104_slave_dsp_communication_init();
/* init and start of 3-phase PWM generation on slave DSP */
// pour les fullPWM dutyx sont initialisé à 0.5
ds1104_slave_dsp_pwm3_init(task_id, MLIperiod, duty1, duty2, duty3, deadband, sync_pos);
ds1104_slave_dsp_pwm3_start(task_id);
```

IM IRFO Indc.C : void main(void)

```
/* registration of PWM duty cycle update command */
ds1104_slave_dsp_pwm3_duty_write_register(task_id, &index);
/* ADC Init and Sync */
ds1104_adc_mux( 1); /* le 16 bit ADC sur channel 1) */
/* 12 bits ADC 1 to 4 ie 2 to 5 are Courants Ias, Ibs et f.e.m. Ealphas et Ebetas */
ds1104_adc_trigger_setup( 1, DS1104_TRIGGER_ENABLE);
ds1104_adc_trigger_setup( 2, DS1104_TRIGGER_ENABLE);
ds1104_adc_trigger_setup( 3, DS1104_TRIGGER_ENABLE);
ds1104_adc_trigger_setup( 4, DS1104_TRIGGER_ENABLE);
ds1104_adc_trigger_setup( 5, DS1104_TRIGGER_ENABLE);
ds1104_syncin_edge_setup( DS1104_SYNC_TRIGGER_RISING);

/* initialization of PWM sync interrupt */
ds1104_set_interrupt_vector(DS1104_INT_SLAVE_DSP_PWM,
                            (DS1104_Int_Handler_Type) &PWM_sync_interrupt, SAVE_REGS_ON);
ds1104_enable_hardware_int(DS1104_INT_SLAVE_DSP_PWM);

RTLIB_INT_ENABLE();
/* Background tasks */
while(1)
  {
  RTLIB_BACKGROUND_SERVICE();                        /* background service */
  }
}
```

IM IRFO Indc.C : PWM_sync_interrupt

```
// interrupt service routine for PWM sync interrupt à 100 us
void PWM_sync_interrupt(void)
{
ds1104_bit_io_set( DS1104_DIO0 | DS1104_DIO1);

RTLIB_TIC_START();                              /* start time measurement */
ds1104_bit_io_clear( DS1104_DIO0);

switch(countCurrent)
     {
     case 0 : RegulI_Control();// à DT = 200 us
              break;
     case 1 : Regul_Vitesse();// à DT = 200 us puis *5 = 1ms
              break;
     }
if (++countCurrent==DS_IReg) countCurrent=0;

host_service( 1, 0);// new host interface at dt=1e-4 s
ds1104_bit_io_clear( DS1104_DIO1);
exec_time = RTLIB_TIC_READ();
}
```

IM IRFO Regul.C : void RegulI_Control()

```
inline void RegulI_Control()
{
Ias = 25*( -0.0018 + ds1104_adc_read_conv( 2));
Ibs = 25*(  0.0022 + ds1104_adc_read_conv( 3));

C22t( Ias, Ibs, &Ialphas, &Ibetas );
CalcPark( thetas, & Sinthetas, & Costhetas);
ParkInv( Ialphas, Ibetas, Sinthetas, Costhetas, &Ids, &Iqs);

Vas = 943.4*( -0.0014 + ds1104_adc_read_conv( 4));
Vbs = 943.4*( -0.0014 + ds1104_adc_read_conv( 5));
Vdc = 943.4*( -0.0014 + ds1104_adc_read_conv( 1));
E_2=Vdc/2.0;// tension du pont / 2
BY_2E_2 = 1.0/Vdc;

Wm=WmMesure;// Wm=WmEst;// Sensorless ?
Calc_thetas(); //update thetas and Phidr
Calc_theta_M_r();

absWm = fabs(Wm);// Defluxage s'il y a lieu
switch ( Deflux_Type )
     {
     case 2 :/* Deflux continu */
     if ( absWm<=WmBaseB ) Idsref=IdsrefBase;
     if ( absWm>WmBase ) Idsref=Kdeflux/absWm;
     break;
     }
```

IM IRFO Regul.C : void RegulI_Control()

```
if ( CurrentControllerType ){/* Controlleur Flou */
     VdsrefMax = E_2/1.414;
     BoucleFloueIds(); /*regul FLOUE de Ids */
     VqsrefMax=sqrt(E_2*E_2-Vdsref*Vdsref);
     BoucleFloueIqs(); /*regul FLOUE de Iqs */
     }
else {/* controlleurs classiques */
     VdsrefMax = E_2/1.414;
     BoucleIds();
     VqsrefMax=sqrt(E_2*E_2-Vdsref*Vdsref);
     BoucleIqs();
     }
// type de Cmde vect ou V/f (independante ou liée)
switch ( FlagVectorControl )
     { ...
     case 2 ://V/fliés par K_V_f
          Vsref=K_V_f*fs;
          Wsref=2*PI*fs;
          Vdsref=Vsref;
          Vqsref=0.0;
          /* limiteur de tension */
          if ( Vdsref>270.0)Vdsref=270.0;
          if ( Vdsref<-270.0) Vdsref=-270.0;
          break;
     default : //case 1 ou autre : Vector control
     break;
     }
```

IM IRFO Regul.C : void RegulI_Control()

```
Park( Vdsref, Vqsref, Sinthetas, Costhetas, &Valphasref, &Vbetasref);
C32( Valphasref, Vbetasref, &Vasref, &Vbsref, &Vcsref );

// protection contre surcourant et bus continu
if (fabs(Ias)>IsmaxProtect || fabs(Ibs)>IsmaxProtect )FlagInhibMLI = 1;
if (Vdc > VdcMax)FlagInhibMLI = 1;

/* write PWM Duty cycle to slave DSP and test for error */
if  (FlagInhibMLI){/* met les tensions à 0 */
     duty1=0.5;
     duty2=0.5;
     duty3=0.5;
     }
else{
     duty1 = -Vasref*BY_2E_2+0.5;
     duty2 = -Vbsref*BY_2E_2+0.5;
     duty3 = -Vcsref*BY_2E_2+0.5;
     }
ds1104_slave_dsp_pwm3_duty_write( task_id, index, duty1, duty2, duty3);
}
```

IM IRFO Regul.C : void Regul_Vitesse()

```
inline void Regul_Vitesse()
{
if ( ++countSpeed!=DS_WReg )return;
countSpeed=0;
// Interruption a 1 ms
inputWm();// La vitesse, Mpos et theta (position du rotor avec reset sur Index)

if ( FlagPosControl )
if ( ++countPos==DS_PReg ){
     countPos=0;
     /* get new mposref automaticly */
     /* regul */
     BouclePosition();
     }
...

switch ( SpeedControllerType )
     {
     case 1 :BoucleVitesseFloue(); break;/* Controlleur Flou */
     default :BoucleVitesse();/*classique */
     }
...
}
```

IM IRFO Regul.C : void inputWm() void Calc_thetas()

```
inline void inputWm()
{
      /* read incremental encoder counter */
    inc_k = ds1104_inc_position_read( CodeurCh, DS1104_INC_LINE_SUBDIV_4);
      /* calculate mechanical speed rd/s */
    FIR_Wm[0] = POStoRD_S*(double)( inc_k - Vit_inc_kOld );
    Vit_inc_kOld = inc_k;

    WmMesure=0.2*(FIR_Wm[0]+FIR_Wm[1]+FIR_Wm[2]+FIR_Wm[3]+FIR_Wm[4]);
    FIR_Wm[4]=FIR_Wm[3];
    FIR_Wm[3]=FIR_Wm[2];
    FIR_Wm[2]=FIR_Wm[1];
    FIR_Wm[1]=FIR_Wm[0];
}
//-----------------------------------------------------------------------------
// calcul de thetas, Wr, Ws
inline void Calc_thetas()
{
    Wr= Iqsref / Tr / Idsref;
    Phidr+=DT*( M*Ids-Phidr)/ Tr;
    Ws = P*Wm + Wr;
    if (FlagVectorControl != 1)  Ws = Wsref;// pas de vector control donc V/f
    thetas += DT*Ws;// 200 us
    if ( thetas > PI)thetas-=TWOPI;
    if ( thetas < -PI)thetas+=TWOPI;
}
```

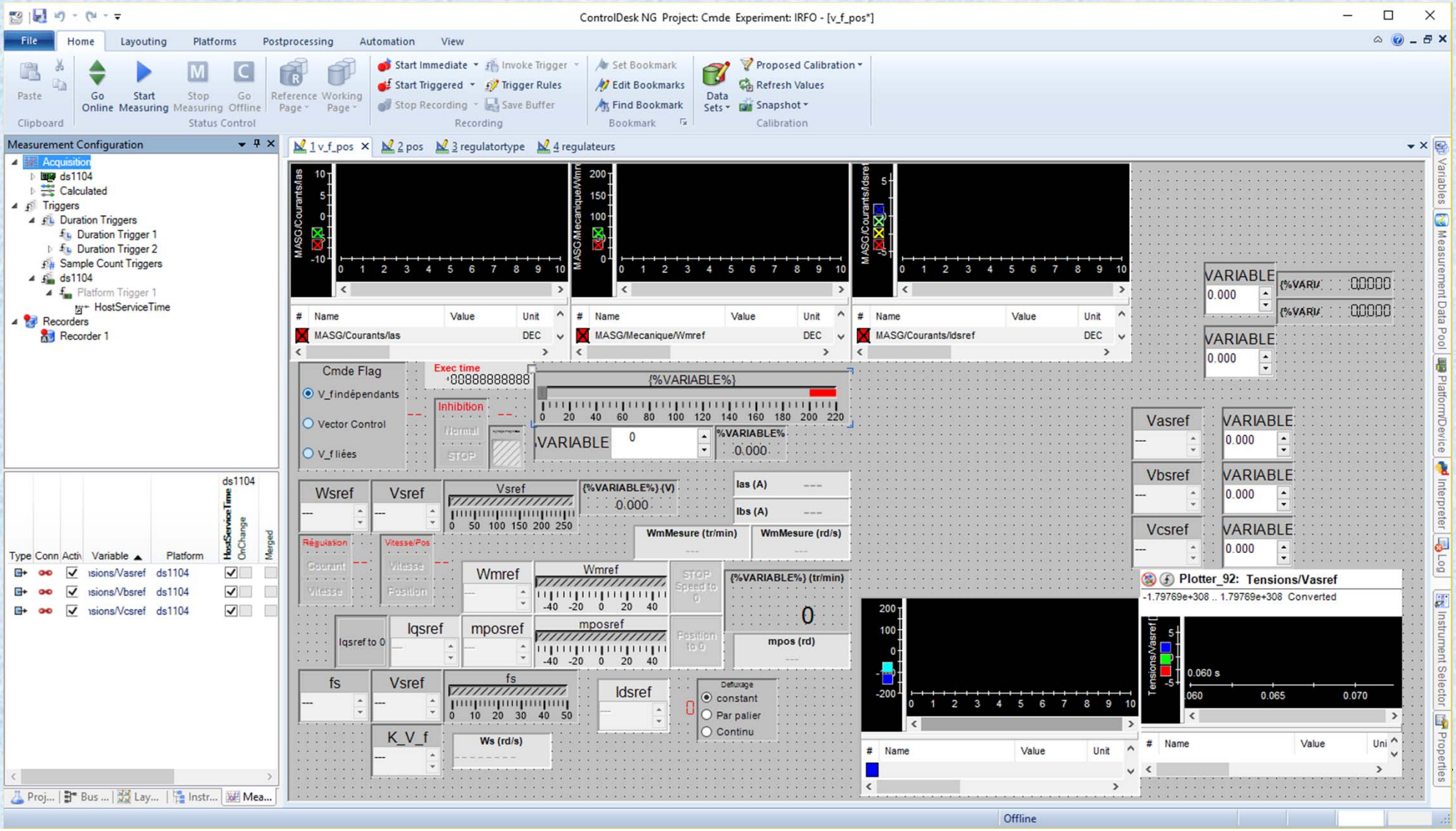
IM IRFO Regul.C : void BoucleIds() void BoucleIqs()

```
inline void BoucleIds()
{
     e = Idsref-Ids;
     Vdsref = Kp_Ids*e + xe_Ids;
     if ( fabs(Vdsref) <= VdsrefMax ) xe_Ids += Ki_Ids*e;
     /* decouplage */
     if (FlagDecouplage) Vdsref-=sigma*Ls*Ws*Iqs;

     /* limiteur de tension */
     if ( Vdsref>VdsrefMax) Vdsref=VdsrefMax;
     if ( Vdsref<-VdsrefMax) Vdsref=-VdsrefMax;
}
//-------------------------------------------------------------------------
inline void BoucleIqs()
{
     e=Iqsref-Iqs;
     Vqsref = Kp_Iqs*e + xe_Iqs;
     if ( fabs(Vqsref)<=VqsrefMax ) xe_Iqs += Ki_Iqs*e;
     /* decouplage */
     if (FlagDecouplage) Vqsref+=sigma*Ls*Ws*Ids+M/Lr*Ws*Phidr;
     /* limiteur de tension */
     if ( Vqsref>VqsrefMax) Vqsref=VqsrefMax;
     if ( Vqsref<-VqsrefMax) Vqsref=-VqsrefMax;
}
```

IM IRFO Regul.C : void BoucleVitesse() / BouclePosition()

```
inline void BoucleVitesse()
{  // regul IP anti Windup
     xeW += KiW / KpW*( Wmref - Wm - KTiW*(OldSortie - OldSortieSat) );
     /* limite le non sature pour eviter tt pb de debordement !!! */
     if ( xeW>xeWMax)xeW=xeWMax;
     if ( xeW<-xeWMax)xeW=-xeWMax;
     OldSortie = KpW*( xeW -Wm);

     OldSortieSat=OldSortie;
     if ( OldSortieSat>IqsrefMax)OldSortieSat=IqsrefMax;
     if ( OldSortieSat<-IqsrefMax)OldSortieSat=-IqsrefMax;
     Iqsref=OldSortieSat;
     /* limiteur de courant et regul IP classique */
     /*
     Iqsref=KpW*(xeW-Wm);
     if ( fabs(Iqsref)<=IqsrefMax )xeW+=KiW/KpW*(Wmref-Wm);
     if ( Iqsref>IqsrefMax)Iqsref=IqsrefMax;
     if ( Iqsref<-IqsrefMax)Iqsref=-IqsrefMax;
     */
} //-------------------------------------------------------------------------------
inline void BouclePosition()
{ // regul P classique
     Wmref= KpP * ( mposref-mpos);
     /* limiteur de vitesse */
     if ( Wmref>WmrefMax) Wmref=WmrefMax;
     if ( Wmref<-WmrefMax) Wmref=-WmrefMax;
}
```

ControlDesk NG IRFO

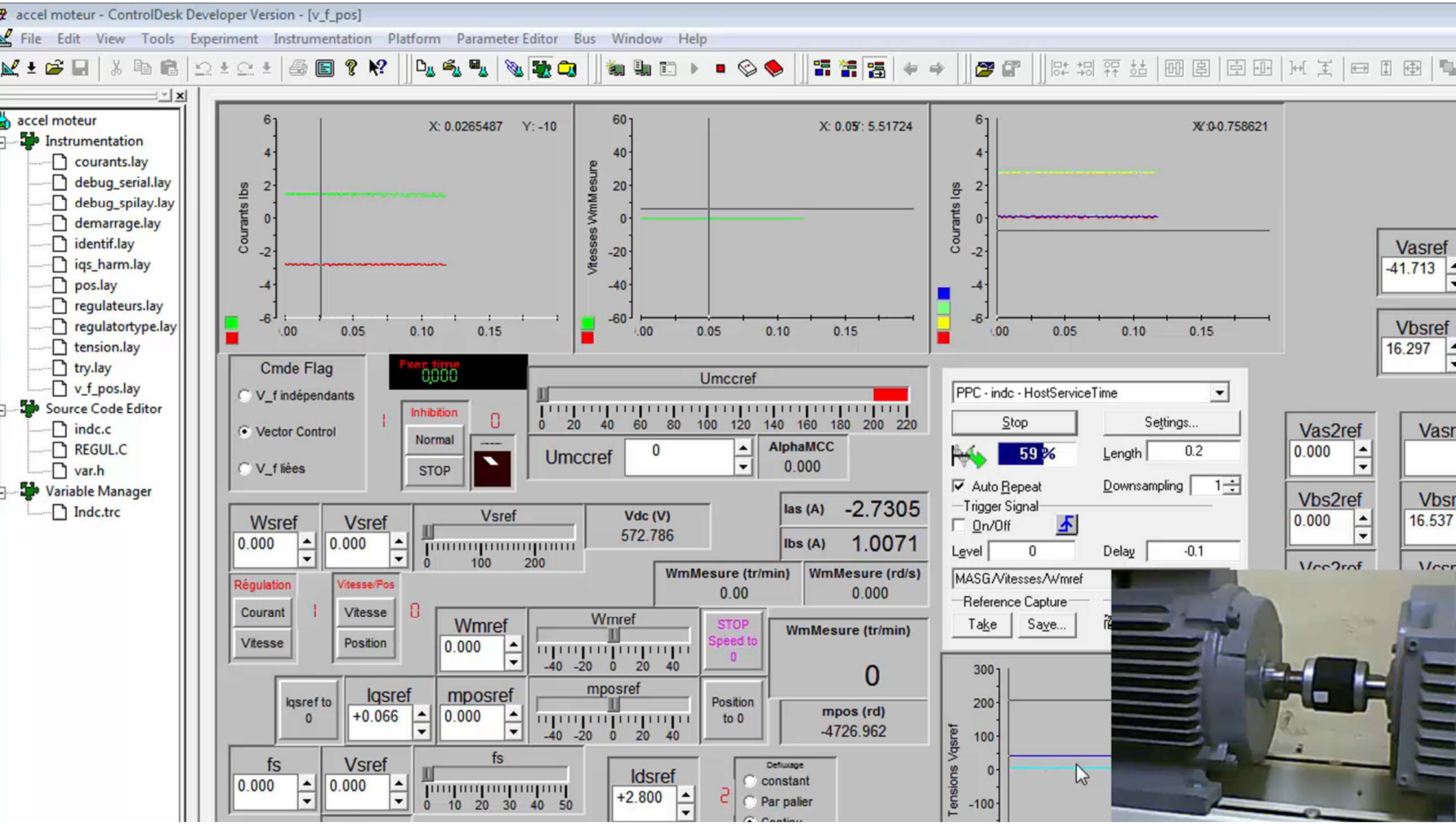


IM IRFO program explanation

https://www.youtube.com/watch?v=R8w7HcxgM70&index=2&list=PLXYd8IyLhtrEpweSyHTwZMjiJzlPIBqAh

IM IRFO experimental results

https://www.youtube.com/watch?v=WOIRo1bIUV0&list=PLXYd8IyLhtrEpweSyHTwZMjiJzlPIBqAh&index=3


accel moteur - ControlDesk Developer Version - [v_f_pos]
File Edit View Tools Experiment Instrumentation Platform Parameter Editor Bus Window Help
accel moteur
Instrumentation
courants.lay
debug_serial.lay
debug_spilay.lay
demarrage.lay
identif.lay
iqs_harm.lay
pos.lay
regulateurs.lay
regulatortype.lay
tension.lay
try.lay
v_f_pos.lay
Source Code Editor
indc.c
REGUL.C
var.h
Variable Manager
Indc.trc
Courants Ibs
Vitesses WmMesure
Courants Iqs
Cmde Flag
V_f indépendants
Vector Control
V_f liées
Exec time
0.000
Inhibition
Normal
STOP
Umccref
AlphaMCC
0.000
Wsref
Vsref
Vdc (V)
572.786
Ias (A) -2.7305
Ibs (A) 1.0071
WmMesure (tr/min)
0.00
WmMesure (rd/s)
0.000
Régulation
Courant
Vitesse
Vitesse/Pos
Vitesse
Position
Wmref
STOP Speed to 0
WmMesure (tr/min)
0
Iqsref to 0
Iqsref
+0.066
mposref
Position to 0
mpos (rd)
-4726.962
fs
Idsref
+2.800
Defluxage
constant
Par palier
PPC - indc - HostServiceTime
Stop
Settings...
59 %
Length 0.2
Auto Repeat
Downsampling 1
Trigger Signal
On/Off
Level 0
Delay -0.1
MASG/Vitesses/Wmref
Reference Capture
Take
Save...
Tensions Vqsref
Vasref
-41.713
Vbsref
16.297
Vas2ref
0.000
Vbs2ref
0.000
16.537

Conclusions

- Presentation of different technologies in EE control
- New offers of solution providers
- Democratization of the use of µC and DSC, DIY, tutorials, Instructables,…
- Massive teaching of µC in college and universities
- Make the right choice, learning curve: Arduino nano → Wemos / ESP8266 / Rasp Pi → PIC24F / dspic 33FJ / 33EP → LaunchPad XL Piccolo / Delfino → dSPACE

References

BOOST-DRV8711 User s Guide slvuai8.pdf
BOOSTXL-DRV8301 Quick Start Guide sldc006.pdf
BOOSTXL-DRV8305EVM User s Guide slvuai8.pdf
BT DM00048919.pdf
curiosity PIC24F 50002534a.pdf
curiosity pub 40001806C.pdf
dspic33FJ128MC802 70291F.pdf
help_mla_tcpip.pdf
help_mla_usb.pdf

LaunchXL-F28027F HW Exp guide spruhh2a.pdf
LaunchXL-F28027F QuickStartGuide sprz413.pdf
LaunchXL-F28069M QuickStartGuide sprui02.pdf
LaunchXL-F28069M User s guide sprui11.pdf
Microchip USB CDC AN 01247a.pdf
MPLAB Xpress 30010119A.pdf
one day workshop sprt517b.pdf

DS1104Features.pdf
DS1104HardwareInstallationConfiguration.pdf
DS1104RTIReference.pdf
DS1104RTLibReference.pdf
dSPACE_DS1104_Catalog2016_E.pdf
Initiation au système dSPACE, de l'interface RTI dans Simulink et de ControlDesk Next Generation 5.4 version 2.0.pdf
ControlDeskNextGenerationBasicPracticesGuide.pdf
ControlDeskNextGenerationMeasurementRecordingTutorial.pdf
ControlDeskNextGenerationMigrationGuide.pdf
ControlDeskNextGenerationReference.pdf

Thank you for your attention

Discussion