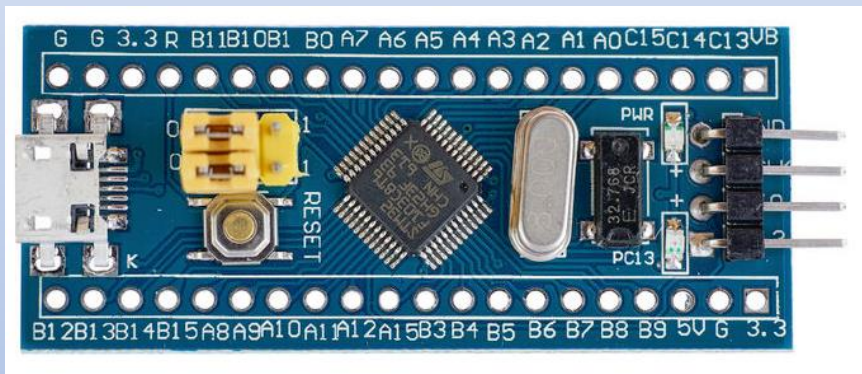


NOTES TECHNIQUES

NT-510 : Gateway STM32 / MySensors / NRF24L01



IOT + DIY = 
MySensors

Sommaire

- 1/ But recherché
- 2/ Le STM 32 & Environnement Arduino
- 3/ NRF24L01
- 4/ MySensors
- 5/ Conclusions

I/ But

Utiliser le package STM32F103 en lieu et place du traditionnel ATmega 325 (quelque soit son package) pour la partie Gateway de l'architecture MySensors utilisée sur DOMO_100.

Rappel : La gateway recoit les événement des capteurs (radio NRF24L01) et revoit les trames reçues sur l'interface série vers le PC qui héberge l'application (Jeedom ou Domoticz)

Avantage de cette solution :

- Configuration 3 à 4 fois plus puissante pour le quasiment le même prix (voir tuto)
 - Dans le cadre de la Gateway cela peut être intéressant (vitesse de traitement)
- Cette configuration existe dans l'environnement Arduino (donc a priori compatible)

US \$1.26

Mémoire flash	: 64ko ou 128ko	(Arduino UNO : 32ko)
Mémoire SRAM	: 20ko	(Arduino UNO : 2ko)
Résolution des entrées	: 4096	(Arduino UNO : 1025)
Fréquence de travail	: 72Mhz	(Arduino UNO : 16Mhz)
Tension de sortie	: 3V3	(Arduino UNO : 5V)

	Blue Pill (STM32F103C8T6)	Arduino Nano (ATmega328P)
Flash memory	64KB	32KB
I/O pins	32	23
PWM pins	15	6
Timer	4	3
ADC pins	10	8
Ext. Interrupt pins	15	2
SPI	2	1
UART	3	1
I2C	2	1
CAN	1	0

Défi à relever

- Quid du Driver SPI (pilote le nrf24l01) et du pin out (différent entre les 2 configurations HW)
- Quid du Driver Série (communique avec le PC par le lien série)
- Quid de la Configuration MySensors (Librairie et sources)
 - changer le pin out NRF24L01
 - passer sur STM32F103 Blue Pill (sous Arduino avec les lib MySensors)
 - utiliser la sortie Série qui va bien

I/ Le STM 32 & Environnement Arduino

Tutoriaux :

<https://www.robot-maker.com/forum/blog/62/entry-68-debuter-avec-la-stm32-f103/>

Wiki : http://wiki.stm32duino.com/index.php?title=Blue_Pill

Pinout : <http://reblog.dk/wordpress/wp-content/uploads/2016/07/The-Generic-STM32F103-Pinout-Diagram.pdf>

Site de ST : <http://www.st.com/en/microcontrollers/stm32-32-bit-arm-cortex-mcus.html?querycriteria=productId=SC1169>

Projet à base de STM : <https://www.hackster...ic-probe-e701d4>

Forum : <http://www.stm32duino.com/>

FAQ : <https://www.stm32duino.com/viewtopic.php?t=2526>

Interessant : <https://www.youtube.com/watch?v=EaZuKRSvwd0>

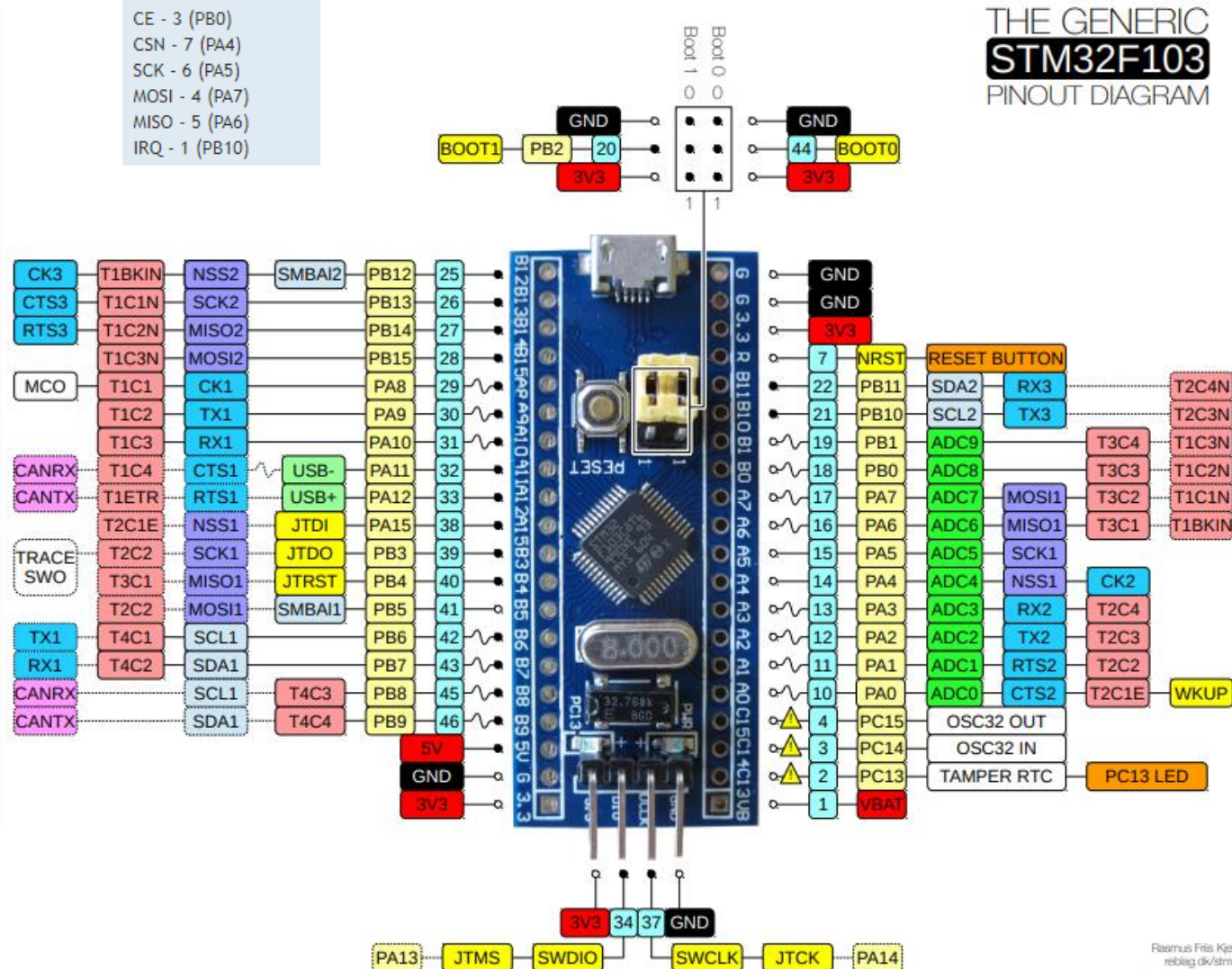
I/ Le STM 32 & Environnement Arduino

Setup:
NRF24 - MAPLE MINI

GND - GND
VCC - VCC
CE - 3 (PB0)
CSN - 7 (PA4)
SCK - 6 (PA5)
MOSI - 4 (PA7)
MISO - 5 (PA6)
IRQ - 1 (PB10)

LEGEND

POWER
GROUND
PHYSICAL PIN
PIN NAME
CONTROL
ANALOG
TIMER & CHANNEL
USART
SPI
I2C
CAN BUS
USB
MISC
BOARD HARDWARE
5V tolerant
Not 5V tolerant
PWM pin
Alternate function
PC13, PC14, PC15: Sink max 3mA, source 0mA, max 2mhz, max 30pF
Absolute MAX 150mA total source/sink for entire CPU
Max ±20mA per pin, ±8mA recommended



I/ Le STM 32 & Environnement Arduino

Coté EDI Arduino : Positionner la bonne carte (éventuellement charger les librairies)

Voir les nombreux tuto sur le sujet

Additional Boards Manager URLs: `http://dan.drown.org/stm32duino/package_STM32duino_index.json`

STM32F1xx/GD32F1xx boards by stm32duino

Boards included in this package:

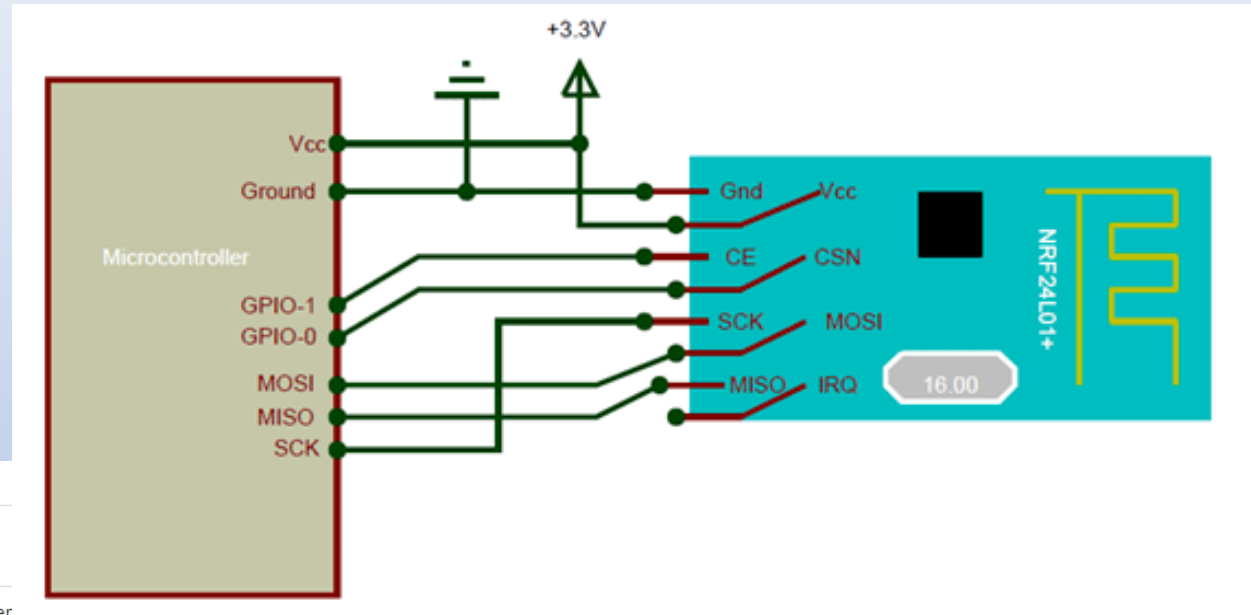
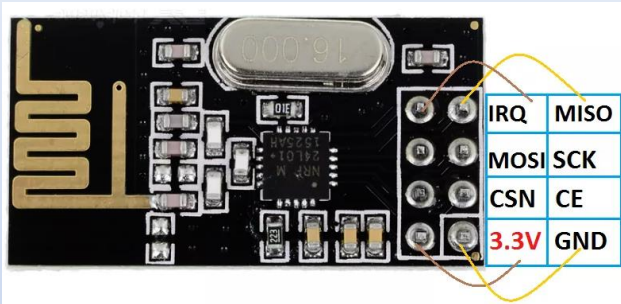
Maple Mini, Maple (Rev 3), Maple (RET6), Microduino Core STM32 to Flash, STM series, Generic STM32F103R series, Generic STM32F103T series, Generic STM32F103C series, GD32F103C series.

[Online help](#)

[More info](#)

III/ NRF24L01

<https://components101.com/wireless/nrf24l01-pinout-features-datasheet>



nRF24L01 Pin Configuration

Pin Number	Pin Name	Abbreviation	Function
1	Ground	Ground	Connected to the Ground of the system.
2	Vcc	Power	Powers the module using 3.3V
3	CE	Chip Enable	Used to enable SPI communication
4	CSN	Chip Select Not	This pin has to be kept high always, else it will disable the SPI
5	SCK	Serial Clock	Provides the clock pulse using which the SPI communication works
6	MOSI	Master Out Slave In	Connected to MOSI pin of MCU, for the module to receive data from the MCU
7	MISO	Master In Slave Out	Connected to MISO pin of MCU, for the module to send data from the MCU
8	IRQ	Interrupt	It is an active low pin and is used only if interrupt is required

https://components101.com/sites/default/files/component_datasheet/nRF24L01%20Datasheet.pdf

III/ NRF24L01

Câblage : gégé

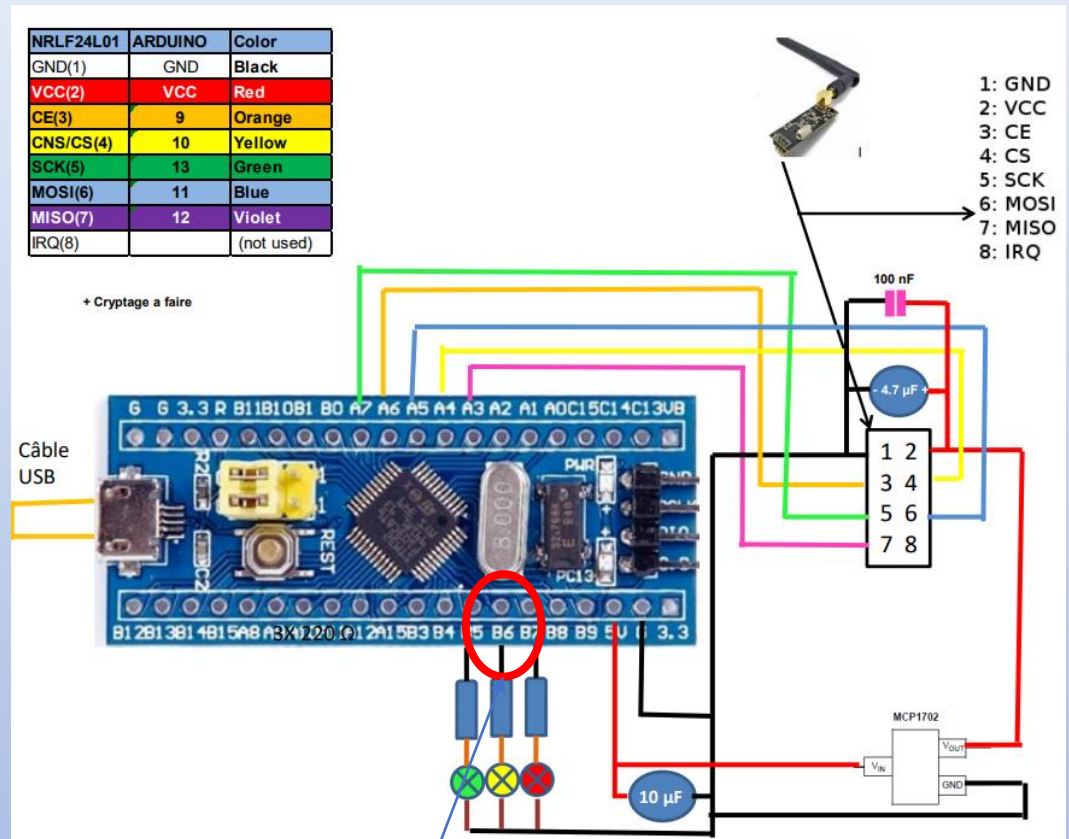
Remarque : Francis

➔ A priori des erreurs dans ton schéma de câblage

Nota : Utiliser directement (à tester)
la sortie 3.3 v du STM32
Évite de rajouter un régulateur

Setup:
NRF24 - MAPLE MINI

GND - GND
VCC - VCC
CE - 3 (PB0)
CSN - 7 (PA4)
SCK - 6 (PA5)
MOSI - 4 (PA7)
MISO - 5 (PA6)
IRQ - 1 (PB10)



Attention à la sortie série rx/tx (probablement pas la sortie câble USB directe)
Les Leds de contrôle ne peuvent pas être ici.

IV/ MySensors

Forum : STM32 / Mysensors : <https://forum.mysensors.org/topic/6519/stm32/73>

Sources (sous git) : <https://github.com/mysensors/MySensors/pull/795>

Des modifications (déjà implémentées plusieurs niveaux dans MySensors sur la version 2.XXX)
-MyConfig.h, MySensors.h, MyHwSTM32F1.cpp/h, MyMainSTM32F1.cpp



Il suffit simplement de positionner la variable la VARIABLE (si nécessaire) :

#define ARDUINO_ARCH_STM32F1 dans le main du programme Gateway (GatewaySerial.ino)
Compiler et télécharger le binaire firmware obtenu.

```
Add STM32F1 HW support

tekka007 committed on 18 Mar 2017

25 ■■■■ MyConfig.h

@@ -529,6 +529,8 @@
529 529 #define MY_RF24_CS_PIN 3
530 530 #elif defined(LINUX_ARCH_RASPBERRYPI)
531 531 #define MY_RF24_CS_PIN 24
532 + #elif defined(ARDUINO_ARCH_STM32F1)
533 + #define MY_RF24_CS_PIN SS
534 534 #else
535 535 #define MY_RF24_CS_PIN 10
536 536 #endif

@@ -679,6 +681,14 @@
679 681 #define MY_RF69_IRQ_PIN RF69_IRQ_PIN
680 682 #endif
681 683
```

```
4 ■■■■ MySensors.h

@@ -66,6 +66,8 @@
66 66 #elif defined(ARDUINO_ARCH_SAMD)
67 67 #include "drivers/extEEPROM/extEEPROM.cpp"
68 68 #include "core/MyHwSAMD.cpp"
69 + #elif defined(ARDUINO_ARCH_STM32F1)
70 + #include "core/MyHwSTM32F1.cpp"
69 71 #elif defined(__linux__)
70 72 #include "core/MyHwLinuxGeneric.cpp"
71 73 #endif
```

```
174 ■■■■ core/MyHwSTM32F1.cpp
```

```
45 ■■■■ core/MyMainSTM32F1.cpp
```

```
69 ■■■■ core/MyHwSTM32F1.h
```

IV/ MySensors

Visiblement déjà fait avec succès

I've successfully tested the RF24 radio with this wiring on a [BluePill](#) board using MySensors 2.2.0-beta:

CLK -> PA5
MISO -> PA6
MOSI -> PA7
CSN -> PA4
CE -> PB0

do not forget to set:

```
#define MY_RF24_CE_PIN PB0
```

Pour de l'aide positioner la variable

#define MY_DEBUG and #define MY_DEBUG_VERBOSE_RF24

Nota : Point qui reste ouvert : l'interface Série / UART :

- Probablement la sortie RX/Tx de la carte PB6/PB7
- Brancher les 2 et voir sur qui cela sort ?.
- Attention Dans le cas de la sortie rx/tx sur la carte : ajouter un convertisseur ftdi (idem pro mini)

V/ Conclusions

A Faire pour la partie Gateway

Extension :

Quid de l'utilisation comme capteur. Le Stm32 semble moins consommer à prouver. Avec infiniment plus de puissance de calcul

https://github.com/freynder/STM32_Sensor

STM32 Sensor Node

A Small STM32F103C8T6 board with all functional pins exposed. Similar to the BluePill board except:

- no regulator. Will be powered by 2 AA batteries.
- no USB
- No LED
- No BOOT0/BOOT1 jumper pins. BOOT0 can be pulled high by a button, BOOT1 is pulled low permanently but exposed by a PIN.
- smaller crystals
- RFM69CW + RFM69(H)W + RFM69HCW footprints + u.FL connector
- VBAT can be disconnected from VCC by cutting the solder jumper on the back.

The size is 36.56mm x 20.32mm. 5 mils trace width and clearance. 20 mils edge clearance.

Tested succesfully V3 with jlcpcb order. Test failed with V3.4 pcbs.io order due to imposed edge clearance. Should be corrected as of V4.3

