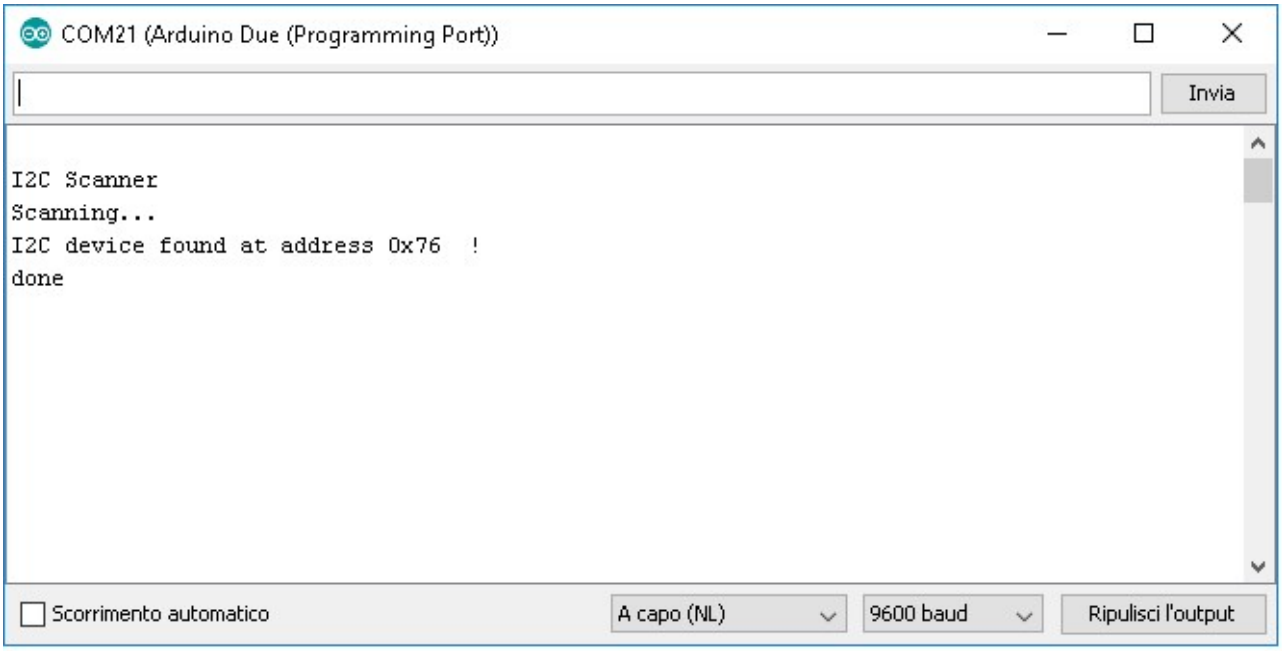
SCHEDA ARDUINO FLIP CLICK + WEATHER CLICK

By Fischetti P.

Lanciare l'utility I2C_Scanner per trovare l'indirizzo della scheda " weather click":



Scaricare e copiare nella dir libraries di arduino i seguenti:

https://github.com/adafruit/Adafruit_BME280_Library

https://github.com/adafruit/Adafruit_Sensor

Modificare il file "Adafruit_BME280.h" alla riga 37 se l'indirizzo riportato da I2C_Scanner e' diverso da 0x77 (come in questo caso)

```
#define BME280_ADDRESS                (0x76)//era  0x77
```

Caricare il seguente sketch:

```
/**************************************************************************
 * Includes
 **************************************************************************/
#include <Wire.h>
#include <SPI.h>
#include <DueTimer.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>

#include "flip_click_defs.h"
```

```
/*****************************************************************************
 * Module Preprocessor Constants
 *****************************************************************************/
// Ambient
#define OPT3001_I2C_ADDR   0x44
#define XBEE_RST A_RST

#define SEALEVELPRESSURE_HPA (1013.25)

/*****************************************************************************
 * Module Preprocessor Macros
 *****************************************************************************/
#define get_temp()          weather_data.temp
#define get_humidity()       weather_data.humidity
#define get_pressure()       weather_data.pressure
#define get_ambient()        weather_data.ambient_light


/*****************************************************************************
 * Module Typedefs
 *****************************************************************************/
typedef struct
{
   float temp;
   float humidity;
   float pressure;
   float ambient_light;
} weather_data_t;

enum
{
   TEMPERATURE = 0,
   PRESSURE,
   HUMIDITY,
   AMBIENT,
   LIGHTNING
};

/*****************************************************************************
 * Module Variable Definitions
 *****************************************************************************/
static weather_data_t weather_data;
#define LEDA 38
#define LEDB 37
#define LEDC 39
#define LEDD 40
byte leds[4] = { LEDA, LEDB, LEDC, LEDD };  /**< Built in LEDs  */

// When to update and transmit
volatile bool update_flag;

Adafruit_BME280 bme;
/*****************************************************************************
```

```
 * Function Prototypes
 ******************************************************************************/
void bme_init( void );
void bme_update( void );

void ambient_init( void );
void ambient_update( void );

void weather_update( void );
void transmit( void );

//ISRs
void update_isr( void );

/*******************************************************************************
 * Function Definitions
 ******************************************************************************/
void setup()
{
  SPI.begin();
  Wire.begin();
  Wire1.begin();
  Serial.begin( 9600 );  // For debugging
  Serial1.begin( 9600 );


  bme_init();

  pinMode( XBEE_RST, OUTPUT );
  digitalWrite( XBEE_RST, HIGH );

  Timer4.attachInterrupt( update_isr ).start( 1000000 );  // Every 1s
}

void loop()
{
  if( update_flag )
  {
     weather_update();
     transmit();
     update_flag = false;
  }
}

void bme_init()
{
  if( !bme.begin( &Wire1))//0x76 ) )
  {
    Serial.println("Could not find a valid BME280 sensor, check wiring!");
    while (1);
  }
}
```

```
void bme_update()
{
   weather_data.temp = bme.readTemperature();
   weather_data.pressure = bme.readPressure() / 100.0f;
   weather_data.humidity = bme.readHumidity();
}

void weather_update()
{
  ambient_update();
  bme_update();
}

void transmit()
{
 char txt[50];
 sprintf( txt, "T:%2.1f,P:%3.0f,H:%2.0f,L:%3.0f", get_temp(), get_pressure(), get_humidity(), get_ambient()
);
 Serial.println( txt );
 Serial1.println( txt );
}


/**
 * @brief Initializes ambient light sensor
 */
void ambient_init()
{
   Wire1.beginTransmission( OPT3001_I2C_ADDR );
   Wire1.write( 0x01 );  /**< Configuration register */
   Wire1.write( 0xC6 );  /**< Set Mode to continuous conversions */
   Wire1.write( 0x10 );
   Wire1.endTransmission();
}

/**
 * @brief Updates ambient light reading from sensor
 */
void ambient_update()
{
   byte tmp[2];
   int i = 0;

   Wire1.beginTransmission( OPT3001_I2C_ADDR ); /**< Issue I2C start signal */
   Wire1.write( 0x00 );
   Wire1.endTransmission();                  /**< stop transmitting */
   Wire1.beginTransmission( OPT3001_I2C_ADDR );
   Wire1.write( 0x00 );
   Wire1.requestFrom( OPT3001_I2C_ADDR, 2 );

   while( Wire1.available() && i != 2 )
      tmp[i++] = Wire1.read();
```

```
   Wire1.endTransmission();

   i = ( ( tmp[0] << 8 ) | tmp[1] );
   tmp[0] = tmp[0] >> 4;
   // Lux equation
   weather_data.ambient_light = 0.01 * ( 2 << tmp[0] ) * i;

   // The most useful ALS range is in the 1 to 1,000 lux range
   if( weather_data.ambient_light >= 1000.0f )
      weather_data.ambient_light = 1000.0f;

   weather_data.ambient_light  /=  10.0f; // Lux to percents conversion
}


void update_isr()
{
   update_flag = true;
}



**********FILE “flip_click_defs.h”:**********
#ifndef FLIP_CLICK_DEFS_H
#define FLIP_CLICK_DEFS_H

/***************
 *** Click D ***
 ***************/
#define D_AN   57
#define D_RST  36
#define D_CS   78
#define D_SCK  76
#define D_MISO 74
#define D_MOSI 75

#define D_PWM  9
#define D_INT  29
#define D_RX   15
#define D_TX   14
#define D_I2C1_SCL 21
#define D_I2C1_SDA 20

/***************
 *** Click C ***
 ***************/
#define C_AN   56
#define C_RST  35
#define C_CS   52
#define C_SCK  76
#define C_MISO 74
#define C_MOSI 75
```

```
#define C_PWM  8
#define C_INT  28
#define C_RX   15
#define C_TX   14
#define C_I2C1_SCL 21
#define C_I2C1_SDA 20

/***************
 *** Click B ***
 ***************/
#define B_AN   55
#define B_RST  34
#define B_CS   4
#define B_SCK  76
#define B_MISO 74
#define B_MOSI 75

#define B_PWM  7
#define B_INT  27
#define B_RX   15
#define B_TX   14
#define B_I2C1_SCL 71
#define B_I2C1_SDA 70

/***************
 *** Click A ***
 ***************/
#define A_AN   54
#define A_RST  33
#define A_CS   77
#define A_SCK  76
#define A_MISO 74
#define A_MOSI 75

#define A_PWM  6
#define A_INT  26
#define A_RX   15
#define A_TX   14
#define A_I2C1_SCL 71
#define A_I2C1_SDA 70

#endif
```

********** RISULTATO ESECUZIONE **********