



Machinechat with STM32F746 and X-NUCLEO-IKS01A3

RobertCNelson Applications Engineer

13d

Software:



- [JEDI One](#)
- [Arduino](#)
- [STM32DUINO](#)

Oct 8

1 / 1

Oct 9

Project Parts:

- [NUCLEO-F746ZG](#)
- [X-NUCLEO-IKS01A3](#)

In Arduino, use this link in the " *Additional Boards Managers URLs* " field: to add STM32 support.

https://github.com/stm32duino/BoardManagerFiles/raw/master/STM32/package_stm_index.json

Add these libraries thru Arduino's Library Manager:

- [STM32Ethernet](#)
- [ArduinoHttpClient](#)
- [ArduinoJson](#)
- [X-NUCLEO-IKS01A3](#)

13d ago



Approve the install of these library dependencies:

- [STM32duino LwIP](#)
- [STM32duino LSM6DSO](#)
- [STM32duino LIS2DW12](#)
- [STM32duino LIS2MDL](#)
- [STM32duino HTS221](#)
- [STM32duino LPS22HH](#)
- [STM32duino STTS751](#)
- [STM32duino LSM6DSOX](#)

Lab1

With the STM32H746 plugged into your PC thru the USB connector (CN1), let's make sure we can program the board by using a simple Blink LED Demo, this code will blink LD1.

```
void setup() {  
    // initialize digital pin LED_BUILTIN as an output  
    pinMode(LED_BUILTIN, OUTPUT);  
};
```



```
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH);
```

```
    delay(500);  
    digitalWrite(LED_BUILTIN, LOW);  
    delay(500);  
}
```

Full Source:

eewiki/machinechat/blob/master/STM32F746_DEVCON_2020/Lab1/Lab1.ino

```
void setup() {  
    // initialize digital pin LED_BUILTIN as an output  
    pinMode(LED_BUILTIN, OUTPUT);  
};  
  
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH);  
    delay(500);  
    digitalWrite(LED_BUILTIN, LOW);  
    delay(500);  
}
```

Verify and Upload:

LD1 should be blinking.

Lab2

The STM32H746 has an onboard ethernet, which can be enabled thru the [STM32Ethernet](#) library. Initialize this library with [Ethernet.begin\(\)](#)., then use [Ethernet.localIP\(\)](#) to print our new IP address to the debug terminal. In this example we should get an IP address from our local DHCP server.

Add LwIP.h and STM32Ethernet.h header files:

```
#include <LwIP.h>  
#include <STM32Ethernet.h>
```

Enable Serial Port for debugging:

```
// Open serial communications and wait for port to open:  
Serial.begin(115200);  
while (!Serial) {  
    ; // wait for serial port to connect. Needed for native USB port only  
}
```

Initialize Ethernet Library

```
// give the ethernet module time to boot up:  
delay(1000);  
  
// start the Ethernet connection:  
Ethernet.begin();  
  
// print the Ethernet board/shield's IP address:  
Serial.print("My IP address: ");  
Serial.println(Ethernet.localIP());
```

Full Source

[eewiki/machinechat/blob/master/STM32F746_DEVCON_2020/Lab2/Lab2.ino](https://eewiki.machinechat/blob/master/STM32F746_DEVCON_2020/Lab2/Lab2.ino)

```
#include <LwIP.h>
#include <STM32Ethernet.h>

void setup() {
  // initialize digital pin LED_BUILTIN as an output
  pinMode(LED_BUILTIN, OUTPUT);

  // Open serial communications and wait for port to open:
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // give the ethernet module time to boot up:
  delay(1000);

  // start the Ethernet connection:
  Ethernet.begin();

  // print the Ethernet board/shield's IP address:
```

This file has been truncated. [show original](#)

Verify and Upload, Arduino's Serial Monitor should report an IP address:

My IP address: 192.168.3.229

Lab3

Next let's verify we have an active link, with [Ethernet.linkStatus\(\)](#), the LED will only be on with an active connection. Test by removing the Ethernet Cable.

```
void loop() {
  if (Ethernet.linkStatus() == LinkON) {
    Serial.println("Link status: On");

    digitalWrite(LED_BUILTIN, HIGH);
  }
  else if (Ethernet.linkStatus() == LinkOFF) {
    Serial.println("Link status: Off");

    digitalWrite(LED_BUILTIN, LOW);
  }
  delay(500);
}
```

Full Source

[eewiki/machinechat/blob/master/STM32F746_DEVCON_2020/Lab3/Lab3.ino](https://eewiki.machinechat/blob/master/STM32F746_DEVCON_2020/Lab3/Lab3.ino)

```
#include <LwIP.h>
#include <STM32Ethernet.h>

void setup() {
  // initialize digital pin LED_BUILTIN as an output
  pinMode(LED_BUILTIN, OUTPUT);

  // Open serial communications and wait for port to open:
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // give the ethernet module time to boot up:
  delay(1000);

  // start the Ethernet connection:
  Ethernet.begin();

  // print the Ethernet board/shield's IP address:
```

This file has been truncated. [show original](#)

Verify and Upload, Arduino's Serial Monitor should also report Link status, matching the LD1, unplug Ethernet Cable to verify.

```
My IP address: 192.168.3.229
Link status: On
Link status: On
Link status: Off
Link status: Off
Link status: On
```

Lab4

Add STTS751 Sensor

```
#include <STTS751Sensor.h>

STTS751Sensor *STTS751_Temp;

// Initialize I2C bus.
Wire.begin();

// Initialize STTS751 Sensor
STTS751_Temp = new STTS751Sensor (&Wire);
STTS751_Temp->Enable();

void loop() {
  //Read STTS751 Temperature
  float STTS751_tempC = 0;
  STTS751_Temp->GetTemperature(&STTS751_tempC);
  float STTS751_tempF = (STTS751_tempC * 1.8) + 32.0F;
```

```
Serial.print(" | Temp[F]: ");  
Serial.print(STTS751_tempF, 2);  
Serial.println(" |");
```

Full Source

[ewiki/machinechat/blob/master/STM32F746_DEVCON_2020/Lab4/Lab4.ino](https://github.com/ewiki/machinechat/blob/master/STM32F746_DEVCON_2020/Lab4/Lab4.ino)



```
#include <LwIP.h>  
#include <STM32Ethernet.h>  
  
#include <STTS751Sensor.h>  
  
STTS751Sensor *STTS751_Temp;  
  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output  
  pinMode(LED_BUILTIN, OUTPUT);  
  
  // Open serial communications and wait for port to open:  
  Serial.begin(115200);  
  while (!Serial) {  
    ; // wait for serial port to connect. Needed for native USB port only  
  }  
  
  // Initialize I2C bus.  
  Wire.begin();
```

This file has been truncated. [show original](#)

Verify and Upload, Arduino's Serial Monitor should now include Temperature data from the STTS751

```
My IP address: 192.168.3.229  
 | Temp[F]: 74.30 |  
Link status: On  
 | Temp[F]: 74.30 |  
Link status: On
```



Lab5

Add HTS221 Sensor

```
#include <HTS221Sensor.h>  
  
HTS221Sensor *HTS221_HumTemp;  
  
// Initialize HTS221 Sensor  
HTS221_HumTemp = new HTS221Sensor (&Wire);  
HTS221_HumTemp->Enable();  
  
// Read HTS221 Humidity and Temperature  
float HTS221_humidity = 0, HTS221_tempC = 0;
```



```

HTS221_HumTemp->GetHumidity(&HTS221_humidity);
HTS221_HumTemp->GetTemperature(&HTS221_tempC);
float HTS221_tempF = (HTS221_tempC * 1.8) + 32.0F;

Serial.print(" | Temp[F]: ");
Serial.print(STTS751_tempF, 2);
Serial.print(" | Temp[F]: ");
Serial.print(HTS221_tempF, 2);
Serial.print(" | Hum[%]: ");
Serial.print(HTS221_humidity, 2);
Serial.println(" |");

```

Full Source

eewiki/machinechat/blob/master/STM32F746_DEVCON_2020/Lab5/Lab5.ino



```

#include <LwIP.h>
#include <STM32Ethernet.h>

#include <STTS751Sensor.h>
#include <HTS221Sensor.h>

STTS751Sensor *STTS751_Temp;
HTS221Sensor *HTS221_HumTemp;

void setup() {
  // initialize digital pin LED_BUILTIN as an output
  pinMode(LED_BUILTIN, OUTPUT);

  // Open serial communications and wait for port to open:
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // Initialize I2C bus.

```

This file has been truncated. [show original](#)

Verify and Upload, Arduino's Serial Monitor should now include Temperature and Humidity data from the HTS221.

```

My IP address: 192.168.3.229
| Temp[F]: 74.07 | Temp[F]: 72.32 | Hum[%]: 41.20 |
Link status: On
| Temp[F]: 74.19 | Temp[F]: 72.32 | Hum[%]: 41.20 |
Link status: On

```



Lab6

Add LPS22HH Sensor

```

#include <LPS22HHSensor.h>

LPS22HHSensor *LPS22HH_PressTemp;

```



```
// Initialize LPS22HH Sensor
LPS22HH_PressTemp= new LPS22HHSensor(&Wire);
LPS22HH_PressTemp->Enable();

// Read LPS22HH Pressure and Temperature.
float LPS22HH_pressure = 0, LPS22HH_tempC = 0;
LPS22HH_PressTemp->GetPressure(&LPS22HH_pressure);
LPS22HH_PressTemp->GetTemperature(&LPS22HH_tempC);
float LPS22HH_tempF = (LPS22HH_tempC * 1.8) + 32.0F;

Serial.print(" | Temp[F]: ");
Serial.print(STTS751_tempF, 2);
Serial.print(" | Temp[F]: ");
Serial.print(HTS221_tempF, 2);
Serial.print(" | Temp[F]: ");
Serial.print(LPS22HH_tempF , 2);
Serial.print(" | Hum[%]: ");
Serial.print(HTS221_humidity, 2);
Serial.print(" | Pres[hPa]: ");
Serial.print(LPS22HH_pressure, 2);
Serial.println(" |");
```

Full Source

eewiki/machinechat/blob/master/STM32F746_DEVCON_2020/Lab6/Lab6.ino

```
#include <LwIP.h>
#include <STM32Ethernet.h>

#include <STTS751Sensor.h>
#include <HTS221Sensor.h>
#include <LPS22HHSensor.h>

STTS751Sensor *STTS751_Temp;
HTS221Sensor *HTS221_HumTemp;
LPS22HHSensor *LPS22HH_PressTemp;

void setup() {
  // initialize digital pin LED_BUILTIN as an output
  pinMode(LED_BUILTIN, OUTPUT);

  // Open serial communications and wait for port to open:
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }
}
```

This file has been truncated. [show original](#)

Verify and Upload, Arduino's Serial Monitor should now include Temperature and Pressure data from the LPS22HH.

My IP address: 192.168.3.229
 | Temp[F]: 73.96 | Temp[F]: 72.14 | Temp[F]: 74.35 | Hum[%]: 41.30 | Pres[hPa]: 979.12 |

Link **status**: On

| Temp[F]: 73.96 | Temp[F]: 72.14 | Temp[F]: 74.35 | Hum[%]: 41.30 | Pres[hPa]: 978.99 |

Link **status**: On

Lab7

Generate Json data object

```
#include <ArduinoJson.h>

// Create a unique ID for the data from each STM32 running this code
const char* jediID = "STM32F7_IKS01A3";

void loop() {
    String postData;

    StaticJsonDocument <200> doc;

    JsonObject context = doc.createNestedObject("context");
    context["target_id"] = String(jediID);

    JsonObject data = doc.createNestedObject("data");
    data["HTS221_humidity"] = HTS221_humidity;
    data["HTS221_tempF"] = HTS221_tempF;
    data["LPS22HH_pressure"] = LPS22HH_pressure;
    data["LPS22HH_tempF"] = LPS22HH_tempF;
    data["STTS751_tempF"] = STTS751_tempF;

    serializeJson(doc, postData);

    //This prints the JSON to the serial monitor screen
    Serial.println(postData);
}
```

Full Source

[eewiki/machinechat/blob/master/STM32F746_DEVCON_2020/Lab7/Lab7.ino](https://forum.digikey.com/t/machinechat-with-stm32f746-and-x-nucleo-iks01a3/9330)

```
#include <LwIP.h>
#include <STM32Ethernet.h>

#include <STTS751Sensor.h>
#include <HTS221Sensor.h>
#include <LPS22HHSensor.h>

#include <ArduinoJson.h>

STTS751Sensor *STTS751_Temp;
HTS221Sensor *HTS221_HumTemp;
LPS22HHSensor *LPS22HH_PressTemp;

// Create a unique ID for the data from each STM32 running this code
const char* jediID = "STM32F7_IKS01A3";

void setup() {
```



```
// initialize digital pin LED_BUILTIN as an output
pinMode(LED_BUILTIN, OUTPUT);
```

This file has been truncated. [show original](#)

Verify and Upload, Arduino's Serial Monitor should now include the Json data object

```
{"context":{"target_id":"STM32F7_IKS01A3"},"data":{"HTS221_humidity":41,"HTS221_tempF":72.
```

Lab8

Transmit Json data object

```
#include <ArduinoHttpClient.h>

char serverAddress[] = "192.168.3.104"; // server address
int port = 8100;

// initialize the library instance:
EthernetClient eth;
HttpClient client = HttpClient(eth, serverAddress, port);

if (Ethernet.linkStatus() == LinkON) {
  Serial.println("Link status: On");

  digitalWrite(LED_BUILTIN, HIGH);

  String contentType = "application/json";

  client.post("/v1/data/mc", contentType, postData);

  // read the status code and body of the response
  int statusCode = client.responseStatusCode();
  String response = client.responseBody();

  Serial.print("Status code: ");
  Serial.println(statusCode);
  Serial.print("Response: ");
  Serial.println(response);
}
else if (Ethernet.linkStatus() == LinkOFF) {
  Serial.println("Link status: Off");

  digitalWrite(LED_BUILTIN, LOW);
}
delay(500);
}
```

Full Source

[eewiki/machinechat/blob/master/STM32F746_DEVCON_2020/Lab8/Lab8.ino](https://forum.digikey.com/t/machinechat-with-stm32f746-and-x-nucleo-iks01a3/9330)

```
#include <LwIP.h>
#include <STM32Ethernet.h>

#include <STTS751Sensor.h>
#include <HTS221Sensor.h>
#include <LPS22HHSensor.h>

#include <ArduinoJson.h>
#include <ArduinoHttpClient.h>

STTS751Sensor *STTS751_Temp;
HTS221Sensor *HTS221_HumTemp;
LPS22HHSensor *LPS22HH_PressTemp;

// Create a unique ID for the data from each STM32 running this code
const char* jediID = "STM32F7_IKS01A3";

char serverAddress[] = "192.168.3.104"; // server address
int port = 8100;
```

This file has been truncated. [show original](#)

Verify and Upload, Arduino's Serial Monitor should now report back a server response code:

Status code: 200

Response: "Data sent successfully"