UNIVERSITÉ
CÔTE D'AZUR

UCA – AIoT Board

Connectivity to External Sensors/Devices

Novembre 26, 2022

Revision 1.0

I. Overview

This document provides information on how to connect external sensors or devices to UCA-AIoT boards. UCA-AIoT hardware summary:

- STM32L476RGT6 (1MB Flash) Micro-controller
- Ublox CAM-M8Q/Quectel L96 GNSS Module
- MicroSD
- Sensors
  - Low-power 9-Axis IMU - TDK InvenSense ICM-20948
  - TVOC & CO2eq Air Quality Sensor - Sensirion SGP30-2.5K
  - MEMS Microphone - Knowles SPH0690LM4H-1
  - [*Optional*] Humidity, Pressure & Temperature sensor - Bosch BME280
  - [*Optional*] Pressure sensor – HopeRF HP203B
  - [*Optional*] Optical Sensor Ambient - Lite-On LTR-303ALS-01
  - [*Optional*] Accelerometers - Kionix KX023-1025-FR

All the information, source code, and demonstration in this document are dedicated to UCA-AIoT boards version **@001 #260621**. The board version can be found on the bottom rear of the board as in Figure 1.

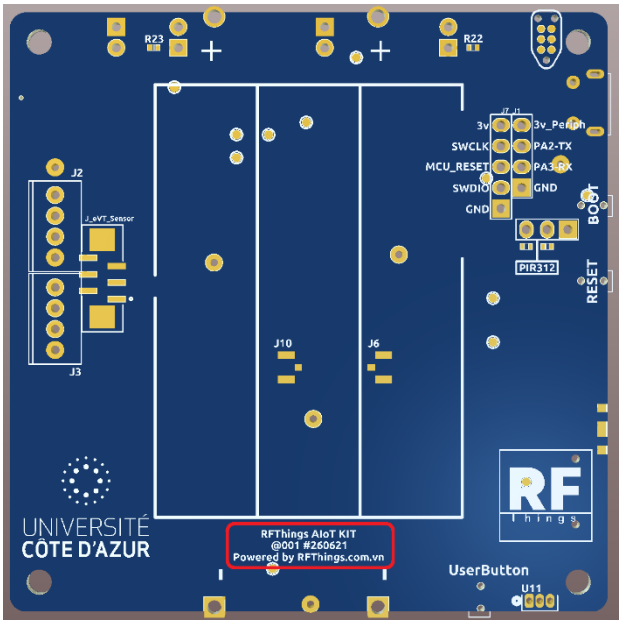

Figure 1. UCA-AIoT Bottom View

Examples are written in Arduino IDE with Arduino Core for UCA AIoT Board. Recommended versions for Arduino IDE and Arduino Core to recompile and upload

example sketches are **1.8.13** and **0.0.4**, respectively. The board schematic, other examples/libraries, and Arduino Core Installation Guide could be found at repository **FabienFerrero/UCA_AIOT**: https://github.com/FabienFerrero/UCA_AIOT

There are J1, J2, J3 headers that could be used to connect external sensors/devices. These headers include pins that could be used as Power supply, ADC, USART, and I2C interfaces. Detailed connections between the MCU and headers are shown in Figure 2.

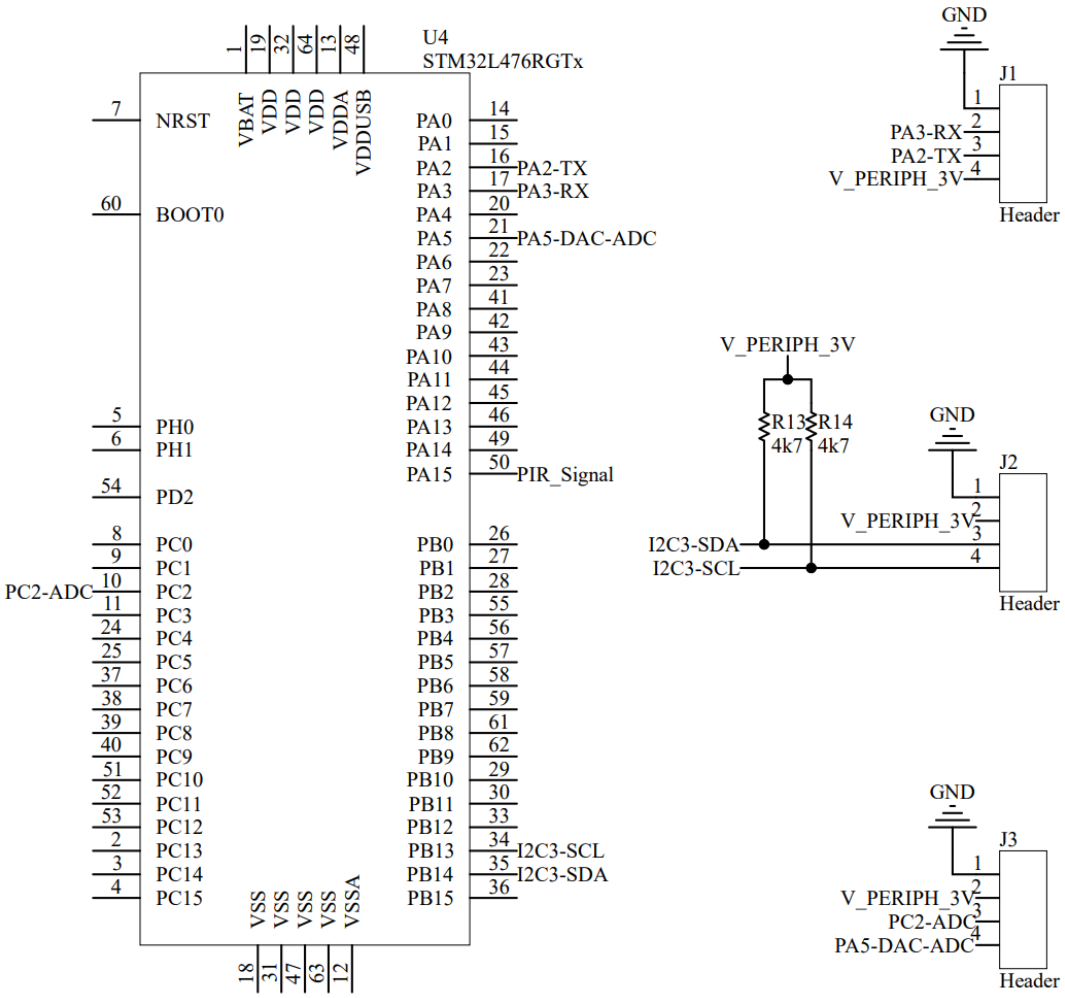

Figure 2. Schematic for J1, J2, J3 headers

Table 1. Function mapping for pins on J1, J2, J3

Pin	Type	GPIOs	ADC	USART	I2C	TIM
J1-1	Ground	-	-	-	-	-
J1-2	I/O	**PA3**	ADC12_IN8	USART2_RX	-	TIM2_CH4 TIM5_CH4 TIM15_CH2
J1-3	I/O	**PA2**	ADC12_IN7	USART2_TX	-	TIM2_CH3 TIM5_CH3 TIM15_CH1
J1-4	VCC_3V	-	-	-	-	-
J2-1	Ground	-	-	-	-	-
J2-2	VCC_3V	-	-	-	-	-
J2-3	I/O	**PB14**	-	-	I2C2_SDA	TIM1_CH2N TIM8_CH2N TIM15_CH1
J2-4	I/O	**PB13**	-	-	I2C2_SCL	TIM1_CH1N TIM15_CH1N
J3-1	Ground	-	-	-	-	-
J3-2	VCC_3V	-	-	-	-	-
J3-3	I/O	**PC2**	ADC123_IN3	-	-	LPTIM1_IN2
J3-4	I/O	**PA5**	ADC12_IN10	-	-	TIM2_CH1 TIM2_ETR TIM8_CH1N LPTIM2_ETR

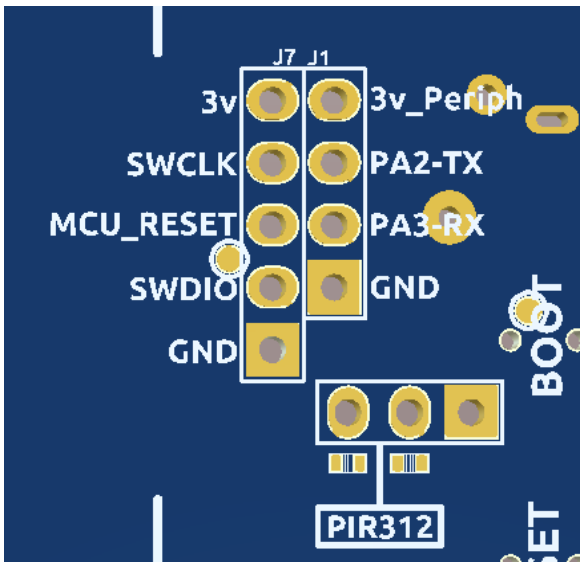

Figure 3. Header J1 position & pin order

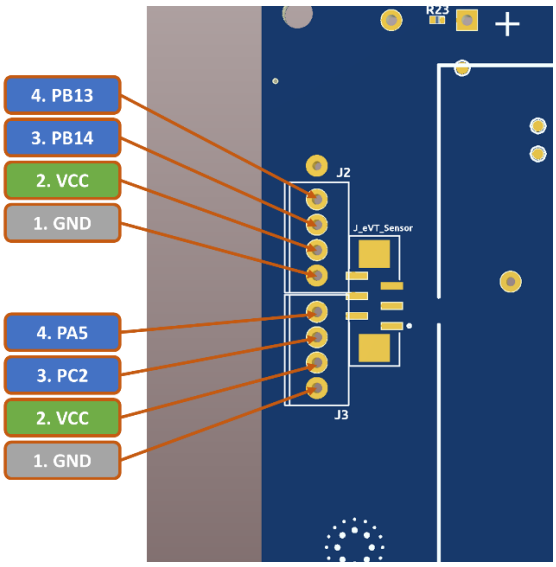

Figure 4. Header J2 & J3 position & pin order

II. Power supply

All power pins on header J1, J2, J3 provide **3V** and connected to a single voltage regulator - U33. U33 is **150mA** low-dropout regulator NCV8170 from ONSEMI. This regulator powers all built-in sensors on UCA – AIOT board. Thus, consider using an external power source to avoid power surge if your sensor(s) requires a high-power consumption or multiple sensors are read at the same time in your applications.

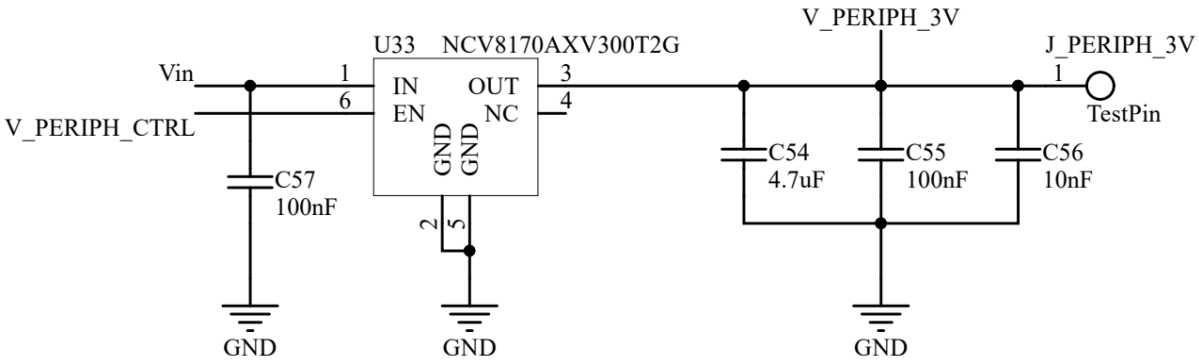

Figure 5. Voltage regulator for built-in sensor and external sensors/devices

III. General-purpose Input/Output (GPIO)

All pins on J1, J2, J3 (except for Power pins) can be configured as GPIOs. Table 2 show correspond pin number in Arduino for these pins. It's fully compatible with Arduino Digital I/O and External Interrupt functions listed below. For detailed information on these functions, refer to **The Arduino Reference text** at https://www.arduino.cc/reference/en/.

```
void pinMode(uint32_t ulPin, uint32_t ulMode);
void digitalWrite(uint32_t ulPin, uint32_t ulVal);
int digitalRead(uint32_t ulPin);

void attachInterrupt(uint32_t pin, voidFuncPtr callback, uint32_t mode);
void detachInterrupt(uint32_t pin);
```

Table 2. Arduino Pin Number

Header	Header Pin Number	MCU Pin Name	Arduino Pin Number
J1	2	PA3	**8**
	3	PA2	**9**
J2	3	PB14	**4**
	4	PB13	**3**
J3	3	PC2	**17**
	4	PA5	**15**

Below source code is an example to toggle **pin J1-2** state every 0,5 seconds:

```
void setup(void) {
  pinMode(8, OUTPUT);
}

void loop(void) {
  digitalWrite(8, HIGH);
  delay(500);
  digitalWrite(8, LOW);
  delay(500);
}
```

IV. Analog to Digital Converter (ADC)

The STM32L4 micro-controller integrated on UCA-AIoT board provides 12-bit successive approximation analog-to-digital converters (ADC). External sensors/devices can be connected to these ADC input via J1 or J3 header. Table 3 shows pin mapping of ADC for Arduino. These pin are fully compatible with Arduino Analog I/O functions.

Default ADC resolution is 10-bit. Function `analogReadResolution(12)` can be used to enable 12-bit resolution. Reference voltage value for ADC is **3V**.

Table 3. ADC function mapping for pins on J1, J3

Pin	Type	GPIOs	ADC	Arduino Digital Pin	Arduino Analog Pin
J1-1	Ground	-	-	-	-
J1-2	I/O	PA3	**ADC12_IN8**	**8**	**A6**
J1-3	I/O	PA2	**ADC12_IN7**	**9**	**A7**
J1-4	VCC_3V	-	-	-	-
J3-1	Ground	-	-	-	-
J3-2	VCC_3V	-	-	-	-
J3-3	I/O	PC2	**ADC123_IN3**	**17**	**A3**
J3-4	I/O	PA5	**ADC12_IN10**	**15**	**A1**

Example for reading 12-bit ADC value on pin **J1-3**:

```
uint16_t adcValue = 0;

void setup(void) {
  analogReadResolution(12);
}

void loop(void) {
  adcValue = analogRead(A7);
  Serial.println(adcValue);
  delay(50);
}
```

V. Universal synchronous receiver transmitter (USART)

UCA-AIoT boards offer a wide range baud rate USART2 to external sensors/devices via **Header J1 – Pin 2 (Rx) / Pin 3 (Tx)**. To access this USART2 in Arduino, use `Serial3` object. `Serial3` is identical to default `Serial` of Arduino in term of function's name, behavior, and interrupt. Below are an `SerialEvent` example that read and echo the input string that end with `'\n'` character.

Table 4. USART function mapping for pins on J1

Pin	Type	GPIOs	USART	Arduino Digital Pin
J1-1	Ground	-	-	-
J1-2	I/O	PA3	**USART2_RX**	**8**
J1-3	I/O	PA2	**USART2_TX**	**9**
J1-4	VCC_3V	-	-	-

```
String inputString = "";      // a String to hold incoming data
bool stringComplete = false;  // whether the string is complete

void setup(void) {
  // initialize serial:
  Serial3.begin(9600);

  // reserve 200 bytes for the inputString:
  inputString.reserve(200);
}

void loop(void) {
  // print the string when a newline arrives:
  if (stringComplete == true) {
    Serial3.println(inputString);

    // clear the string:
    inputString = "";
    stringComplete = false;
  }
}

/*
  SerialEvent occurs whenever a new data comes in the hardware serial
  RX. This routine is run between each time loop() runs, so using delay
  inside loop can delay response. Multiple bytes of data may be
  available.
*/
```

```
void serialEvent() {
  while (Serial3.available()) {
    // get the new byte:
    char inChar = (char)Serial3.read();

    // add it to the inputString:
    inputString += inChar;

     // if the incoming character is a newline, set a flag so the main
loop can do something about it:
    if (inChar == '\n') {
      stringComplete = true;
    }
  }
}
```

VI. Inter-integrated circuit (I2C) interface

The I2C sensors/devices can be communicated to UCA-AIoT board via I2C2 or `Wire2` object in Arduino. I2C2 SDA and SCL lines are connected to Header J2 – Pin 3 and Pin 4, respectively. These **SDA and SCL lines for external sensors/devices are pulled up to VCC_3V by a pair of 4.7 kOhm R13, R14 by default** as shown in Figure 2. If your sensors/devices work at a different logic voltage level (for example 5V), consult to the manufacture to remove, or replace these resistors.

Table 5. I2C function mapping for pins on J2

Pin	Type	GPIOs	I2C	Arduino Digital Pin
J2-1	Ground	-	-	-
J2-2	VCC_3V	-	-	-
J2-3	I/O	PB14	**I2C2_SDA**	**4**
J2-4	I/O	PB13	**I2C2_SCL**	**3**

The `Wire2` that mapped to I2C2 are fully compatible with default Arduino Wire object. The `Wire2` member function used to controller I2C2 are listed below:

```
void begin();
void begin(uint8_t address);
void end();


void beginTransmission(uint8_t);
uint8_t endTransmission(bool stopBit = true);

uint8_t requestFrom(uint8_t address, size_t quantity, bool stopBit =
true);

size_t write(uint8_t data);
size_t write(const uint8_t *buffer, size_t quantity);

int available(void);
int read(void);

void setClock(uint32_t);

void onReceive(void(*callback)(int));
void onRequest(void(*callback)(void));
```

This is an example of device addresses scanning for external I2C sensors connected to UCA-AIoT board via Header J2:

```
#include <Wire.h>

void setup()
{
  Wire2.begin();

  Serial.begin(115200);
  while (!Serial);             // Leonardo: wait for serial monitor
  Serial.println("\nI2C Scanner");

  pinMode(LS_GPS_ENABLE, OUTPUT);
  digitalWrite(LS_GPS_ENABLE, HIGH);
  pinMode(LS_GPS_V_BCKP, OUTPUT);
  digitalWrite(LS_GPS_V_BCKP, HIGH);
  pinMode(SD_ON_OFF, OUTPUT);
  digitalWrite(SD_ON_OFF, HIGH);

  delay(500);
}


void loop()
{
  byte error, address;
  int nDevices;

  Serial.println("Scanning...");

  nDevices = 0;
  for(address = 1; address < 127; address++ )
  {
    // The i2c_scanner uses the return value of
    // the Write.endTransmisstion to see if
    // a device did acknowledge to the address.
    Wire2.beginTransmission(address);
    error = Wire2.endTransmission();

    if (error == 0)
    {
      Serial.print("I2C device found at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.print(address,HEX);
      Serial.println("  !");
```

```
      nDevices++;
    }
    else if (error==4)
    {
      Serial.print("Unknown error at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.println(address,HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");

  delay(5000);  // wait 5 seconds for next scan
}
```

For detail information & more examples on Arduino Wire, refer to: https://www.arduino.cc/reference/en/language/functions/communication/wire/

VII. Others

In additional to above connectivity, UCA-AIoT board also offer a connector/header to directly connect a PIR Motion Sensor and a FFC connector to daughter boards.

1. PIR Motion Sensor AM312

Figure 6. PIR Motion Sensor AM312

Figure 6 shows the PIR Motion Sensor AM312. This is a mini motion sensor with analog voltage output. Table 6 gives detailed information of connection between the sensors and micro-controller STM32L4.

Table 6. Pin map for PIR Motion Sensor AM312

AM312 Pin	MCU Pin	Arduino Analog/Digital Pin
VCC	PA10	**26**
VOUT	PA15	**41**
GND	Ground	-

This is an example of reading motion sensor ADC value:

```
#define PIR_EN_PIN 26
#define PIR_OUT_PIN 41

uint16_t adcValue = 0;

void setup(void) {
  pinMode(PIR_EN_PIN, OUTPUT);
  digitalWrite(PIR_EN_PIN, HIGH); // Power ON
  analogReadResolution(12);
}

void loop(void) {
  adcValue = analogRead(PIR_OUT_PIN);

  Serial.print("Motion Sensor ADC: ");
  Serial.println(adcValue);

  delay(50);
```

```
}
```

2. FFC Connector for daughter board

UCA-AIoT boards offer a 6-pin FFC connector include 3V power supply, 2 x ADC lines, 1 x I2C line. Figure 7 shows the actual connector mounted on boards. For the pin orders of this FFC Connector, refer to Figure 8.

I/O connections on this FFC Connector are sharing with J1, J2, J3 pin that have the same wire names. Ensure that there is no conflict between your external sensors/devices.

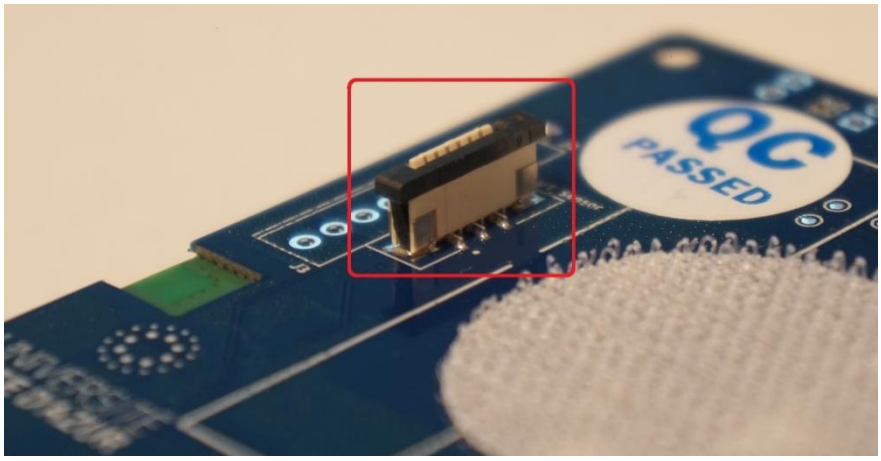

Figure 7. FFC Connector on UCA-AIoT board

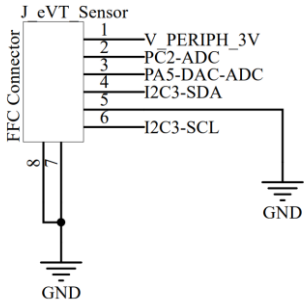

Figure 8. FFC Connector pin order

END