SONY

Start guide and tutorial of Spresense programming with Arduino IDE

R&D Center Tokyo Laboratory 14
Kentaro Matsuura
Satoru Iwasaki

Agenda

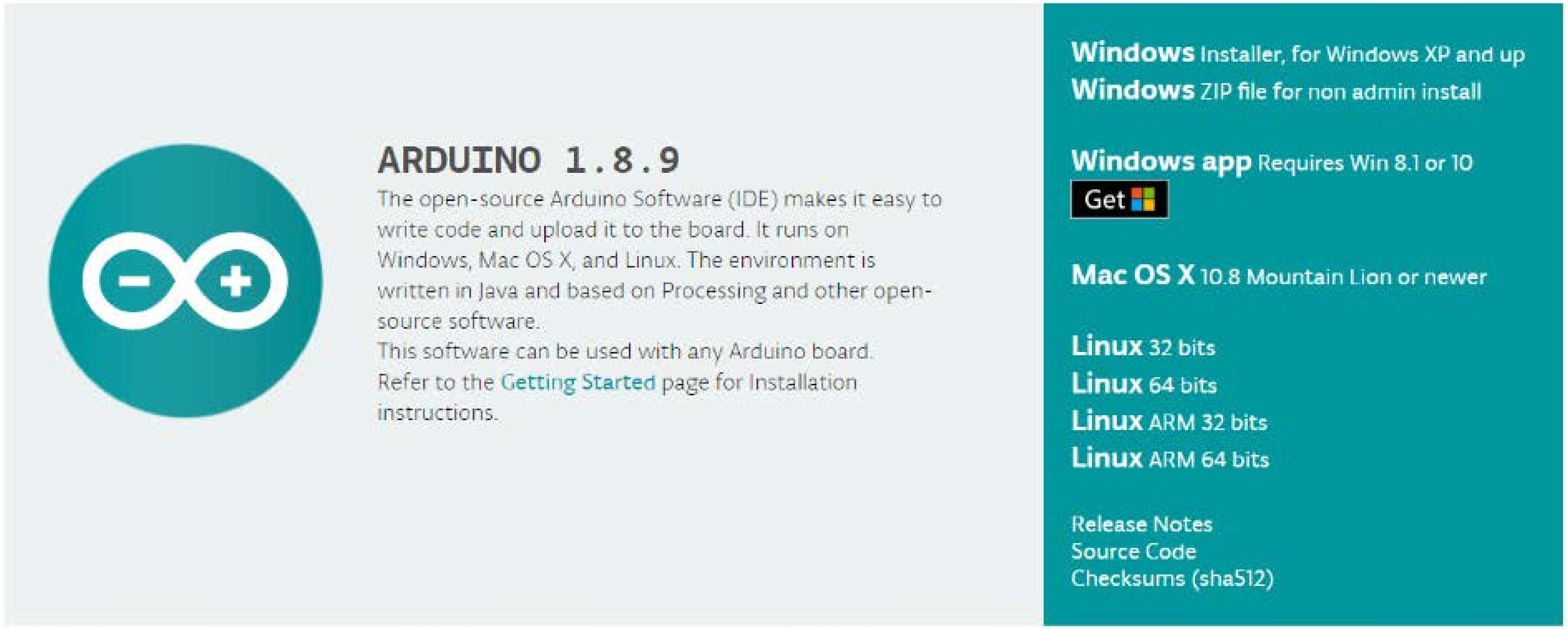
- Start guide
  - Install Arduino IDE
  - Install driver for Spresense
  - Install Spresense Arduino Library
  - Connect Spresense to you PC
  - Install Bootloader
  - Write your own “sketch” on Arduino IDE

- Tutorial
  - How to use GNSS in SPRESENSE
  - How to get the disaster information from QZSS in SPRESENSE
  - Play the recoded sound

Start guide

- Install Arduino IDE
- Install driver for Spresense
- Install Spresense Arduino Library
- Connect Spresense to you PC
- Install Bootloader
- Write your own “sketch” on Arduino IDE

Install Arduino IDE


ARDUINO 1.8.9

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for Installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10

Mac OS X 10.8 Mountain Lion or newer

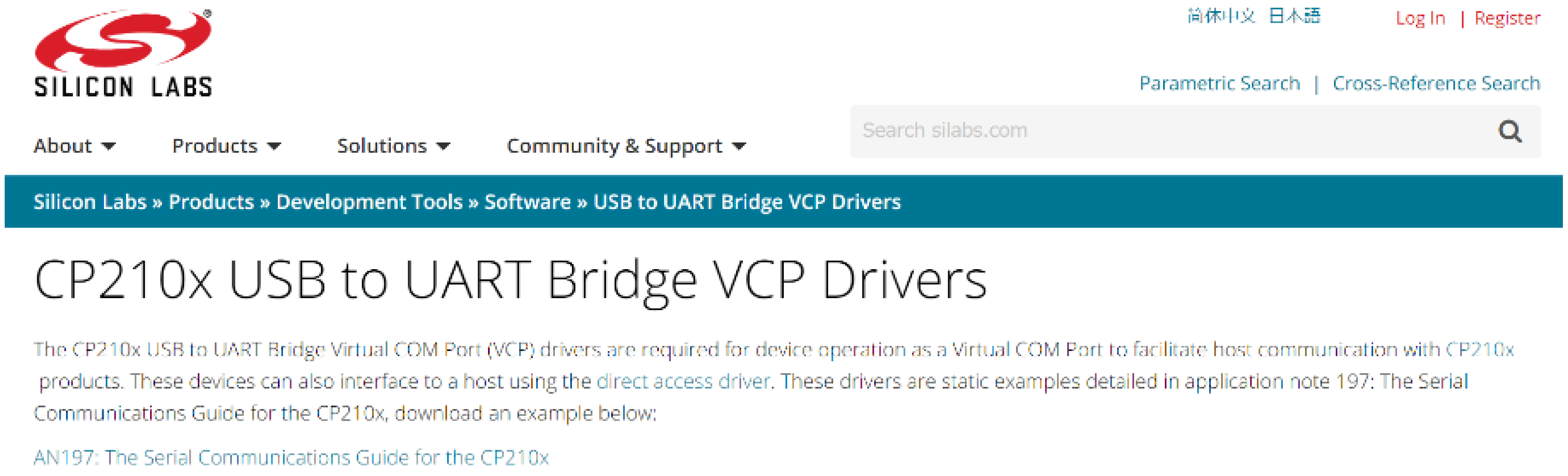
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Release Notes
Source Code
Checksums (sha512)

https://www.arduino.cc/en/main/software#

Install driver for Spresense

https://developer.sony.com/develop/spresense/docs/arduino_set_up_en.html



Silicon Labs » Products » Development Tools » Software » USB to UART Bridge VCP Drivers

CP210x USB to UART Bridge VCP Drivers

The CP210x USB to UART Bridge Virtual COM Port (VCP) drivers are required for device operation as a Virtual COM Port to facilitate host communication with CP210x products. These devices can also interface to a host using the direct access driver. These drivers are static examples detailed in application note 197: The Serial Communications Guide for the CP210x, download an example below:

AN197: The Serial Communications Guide for the CP210x

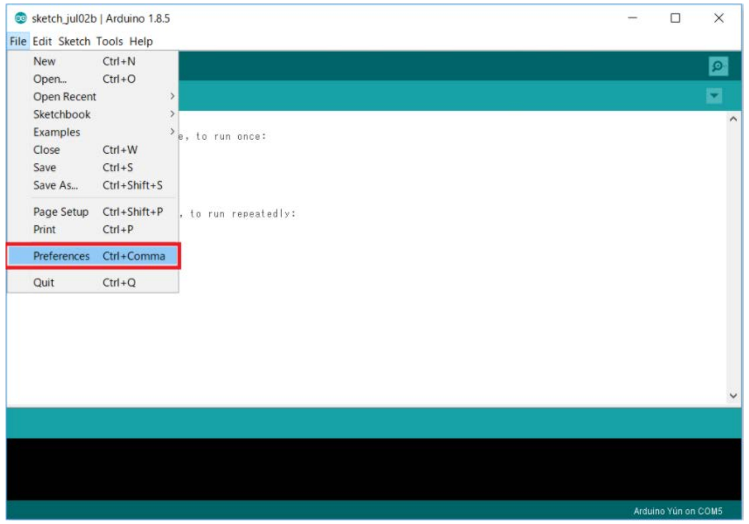
CP210x USB to serial driver for Windows 7/8/8.1
CP210x USB to serial driver (v10.1.3) for Windows 10
CP210x USB to serial driver for Macintosh OSX

If you want to get the latest version, visit the silicon labs site directly

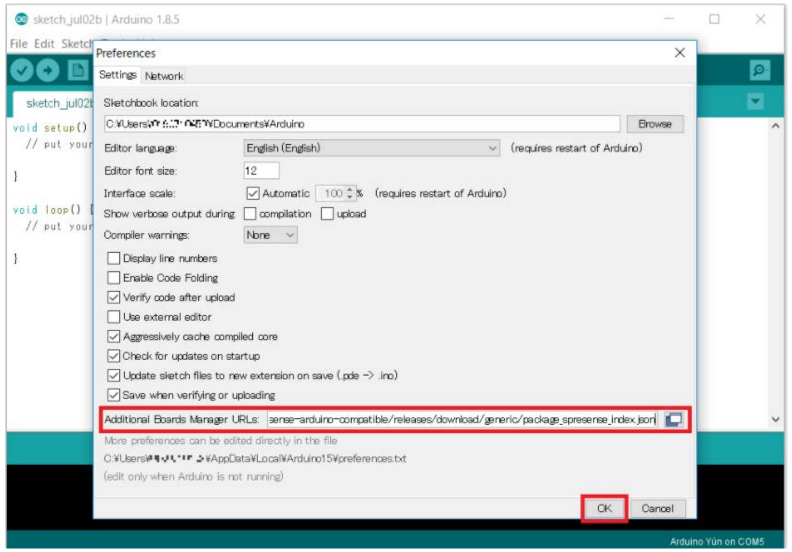
Install Spresense Arduino Library

https://developer.sony.com/develop/spresense/docs/arduino_set_up_en.html

STEP 1
Open Preferences



STEP 2
Copy and paste the following URL into the field called Additional Boards Managers URLs:
https://github.com/sonydevworld/spresense-arduino-compatible/releases/download/generic/package_spresense_index.json



Sketchbook location:
C:¥Users¥[redacted]¥Documents¥Arduino

Editor language: English (English) (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Show verbose output during ☐ compilation ☐ upload

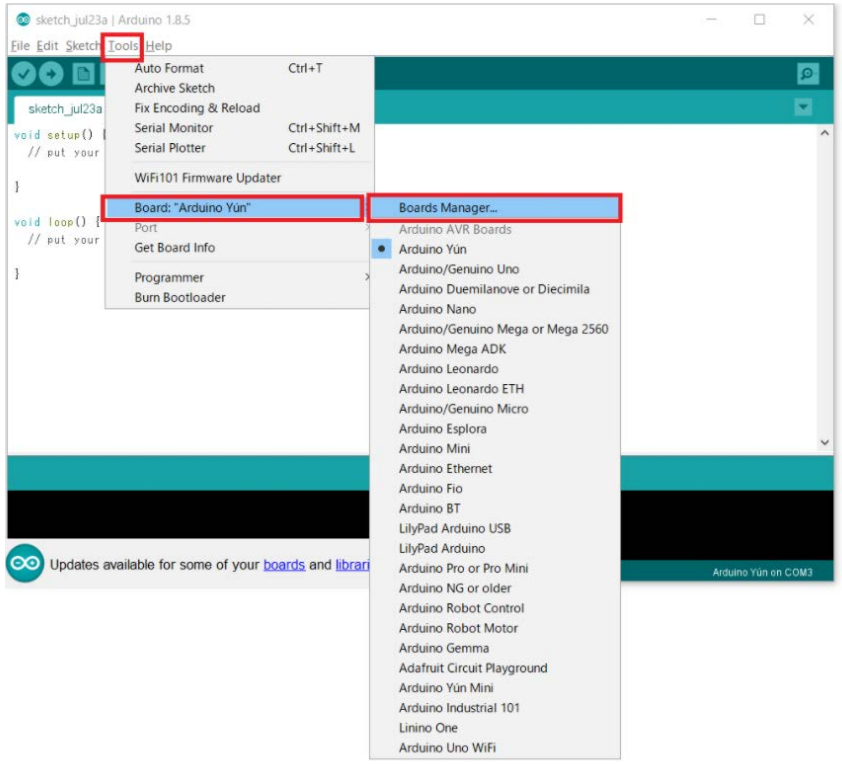
Compiler warnings: None

☐ Display line numbers
☐ Enable Code Folding
☑ Verify code after upload
☐ Use external editor
☑ Aggressively cache compiled core
☑ Check for updates on startup
☑ Update sketch files to new extension on save (.pde -> .ino)
☑ Save when verifying or uploading

Additional Boards Manager URLs: sense-arduino-compatible/releases/download/generic/package_spresense_index.json

More preferences can be edited directly in the file
C:¥Users¥[redacted]¥AppData¥Local¥Arduino15¥preferences.txt
(edit only when Arduino is not running)

STEP 3
Open Boards Manager



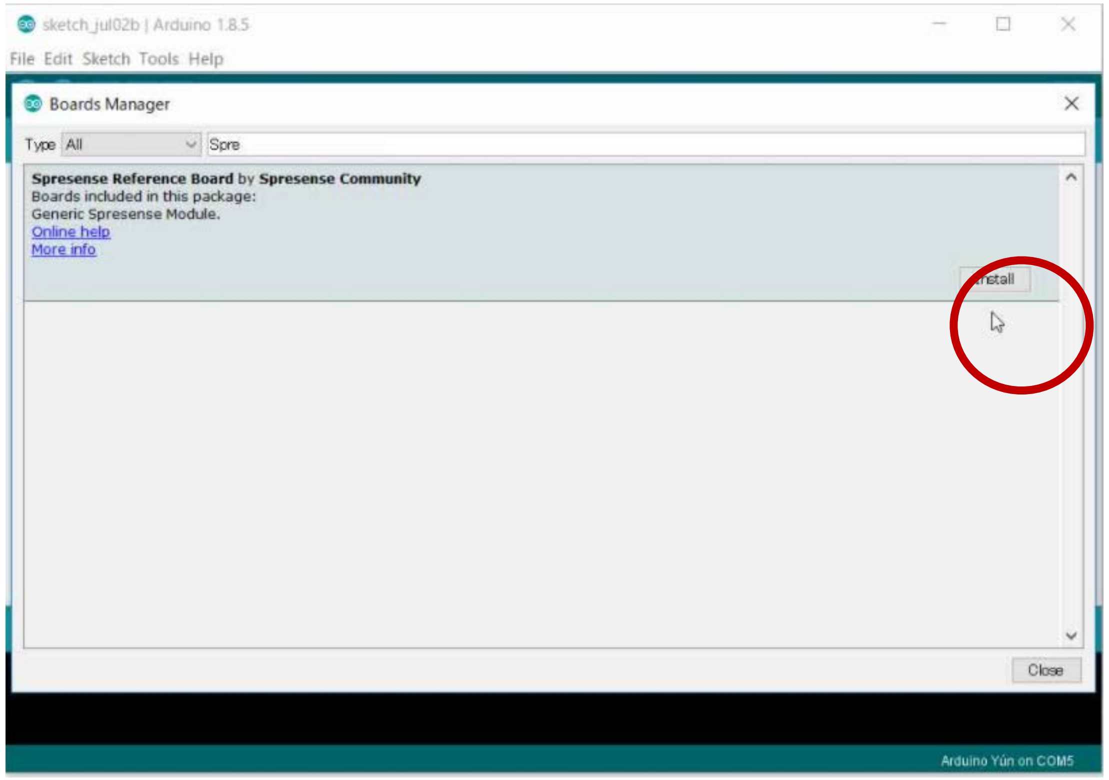
※ Proxy settings is at “Network” tab on “Preference” window

Install Spresense Arduino Library

https://developer.sony.com/develop/spresense/docs/arduino_set_up_en.html

STEP 4

Search for Spresense and select it and click install



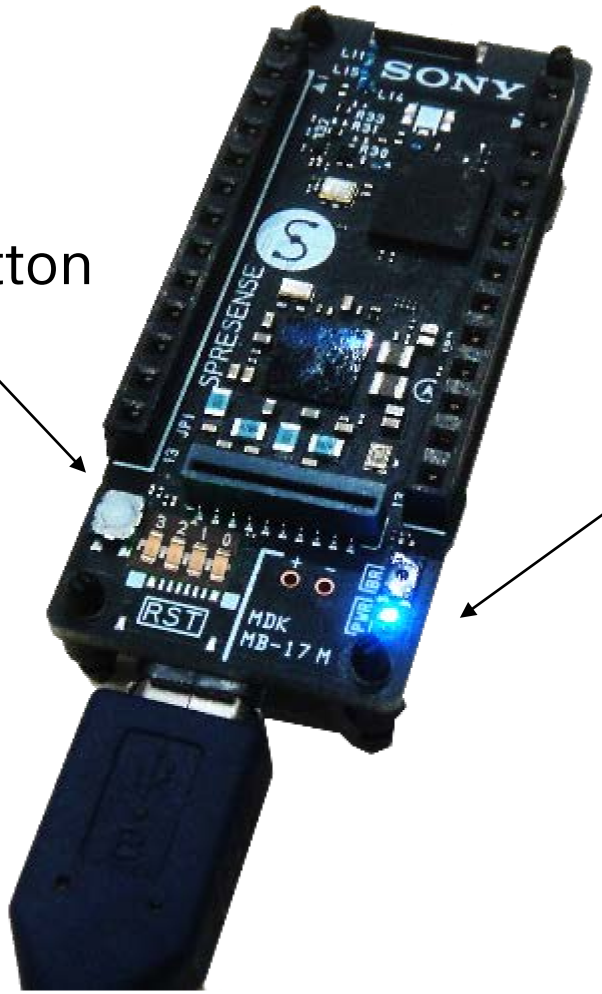
Spresense Reference Board by **Spresense Community**
Boards included in this package:
Generic Spresense Module.
Online help
More info

Connect Spresense to you PC

Reset Button

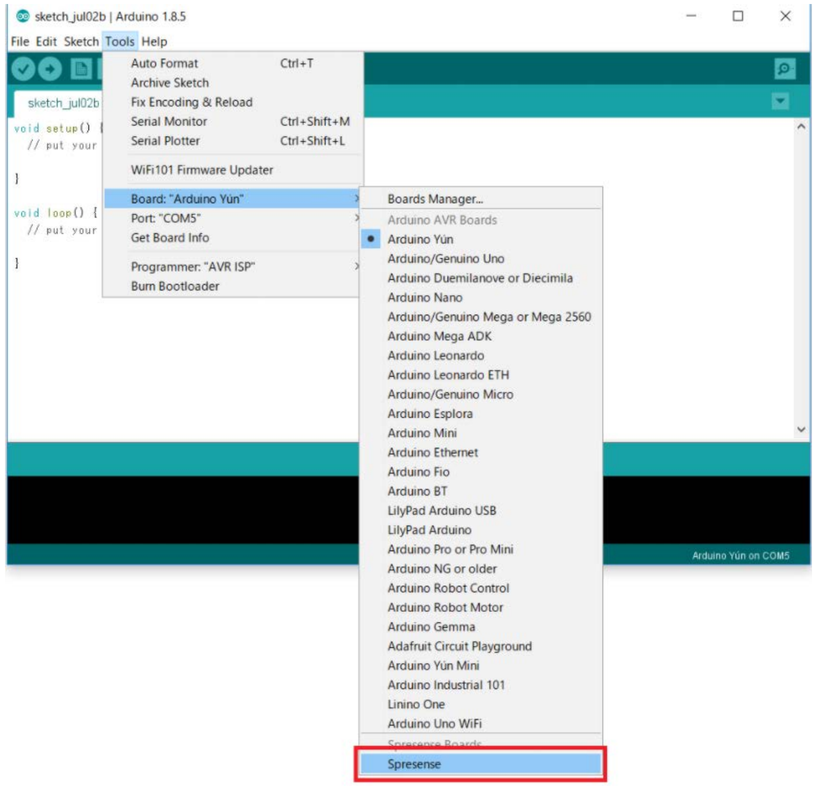
Connect Spresense to your PC with micro USB type B Cable having the capability of data transfer

If you can see a blue light of LED here, the connection is established.

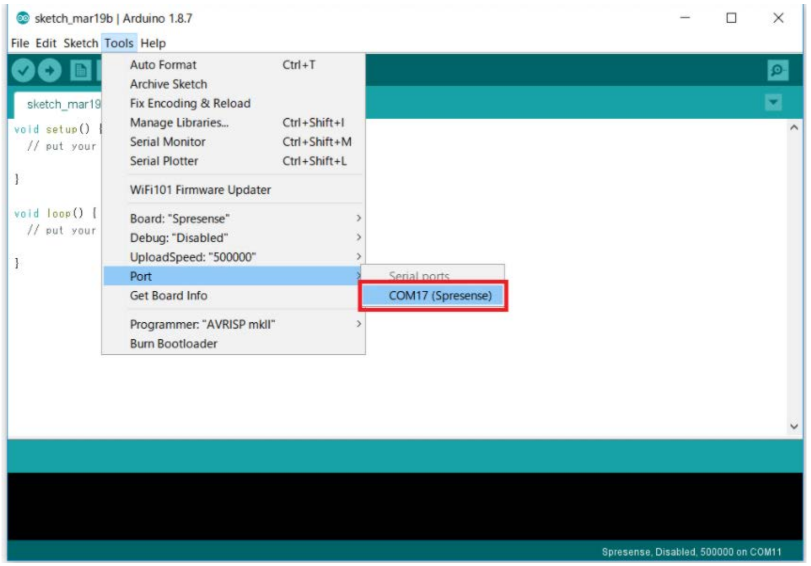


Install Bootloader

STEP 1
Select Spresense board

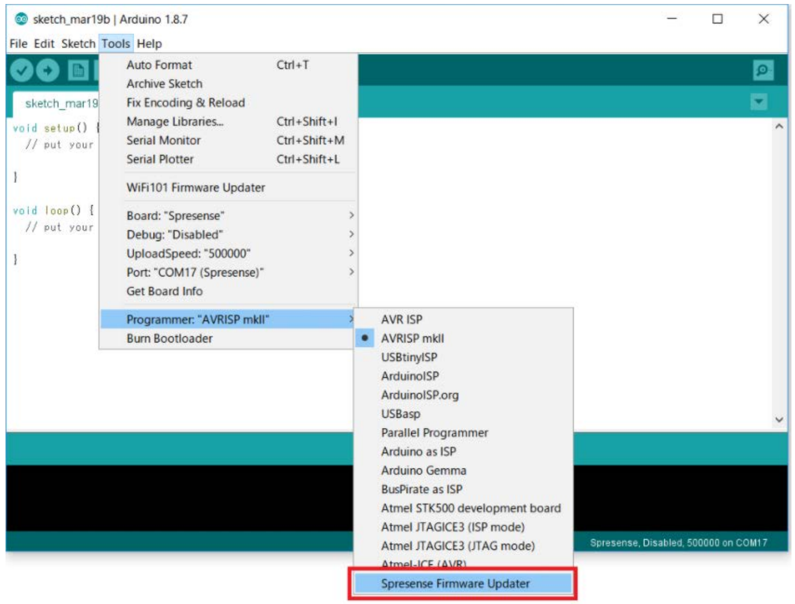


STEP 2
Select the serial port



sketch_mar19b | Arduino 1.8.7
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Spresense"
Debug: "Disabled"
UploadSpeed: "500000"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM17 (Spresense)
Spresense, Disabled, 500000 on COM11

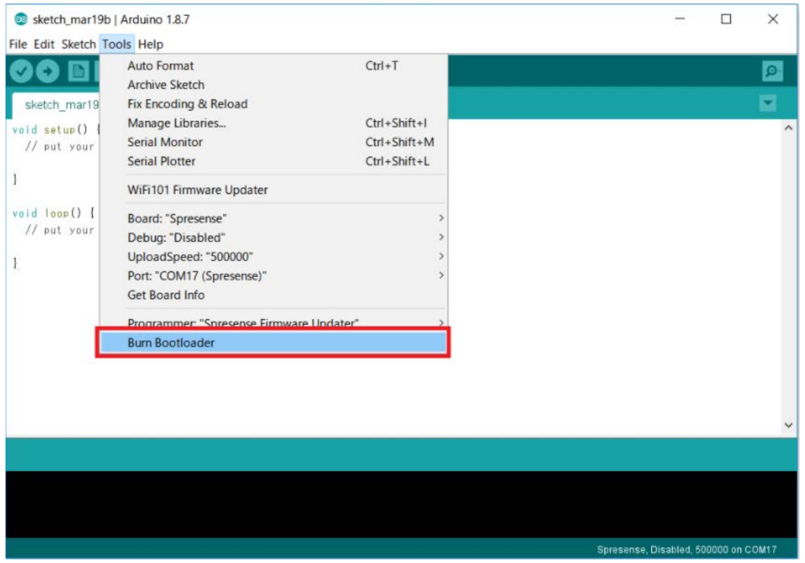
STEP 3
Select Spresense Firmware Updater



sketch_mar19b | Arduino 1.8.7
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Spresense"
Debug: "Disabled"
UploadSpeed: "500000"
Port: "COM17 (Spresense)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
AVR ISP
AVRISP mkII
USBtinyISP
ArduinoISP
ArduinoISP.org
USBasp
Parallel Programmer
Arduino as ISP
Arduino Gemma
BusPirate as ISP
Atmel STK500 development board
Atmel JTAGICE3 (ISP mode)
Atmel JTAGICE3 (JTAG mode)
Atmel-ICE (AVR)
Spresense Firmware Updater
Spresense, Disabled, 500000 on COM17

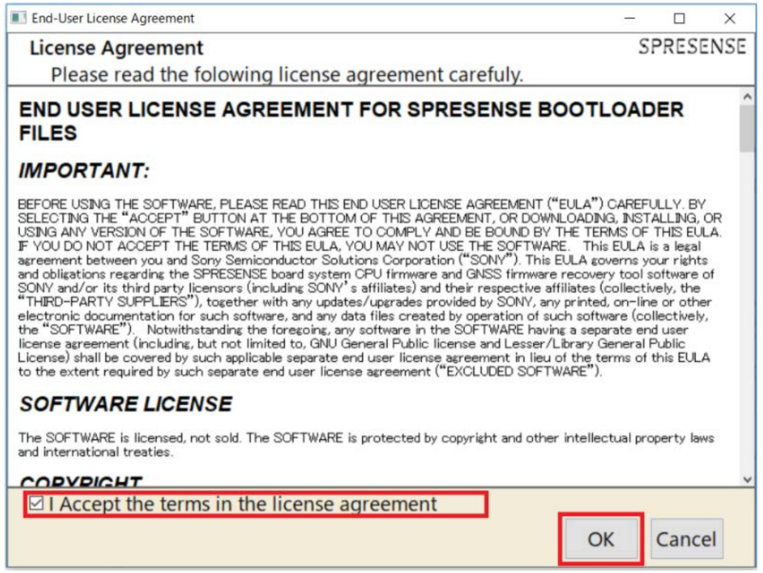
Install Bootloader

STEP 4
Select “Burn Bootloader”



sketch_mar19b | Arduino 1.8.7
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Board: "Spresense"
Debug: "Disabled"
UploadSpeed: "500000"
Port: "COM17 (Spresense)"
Get Board Info
Programmer: "Spresense Firmware Updater"
Burn Bootloader
Spresense, Disabled, 500000 on COM17

STEP 5
Review and accept the EULA by following the dialog



License Agreement

Please read the folowing license agreement carefuly.

END USER LICENSE AGREEMENT FOR SPRESENSE BOOTLOADER FILES

IMPORTANT:

BEFORE USING THE SOFTWARE, PLEASE READ THIS END USER LICENSE AGREEMENT (“EULA”) CAREFULLY. BY SELECTING THE “ACCEPT” BUTTON AT THE BOTTOM OF THIS AGREEMENT, OR DOWNLOADING, INSTALLING, OR USING ANY VERSION OF THE SOFTWARE, YOU AGREE TO COMPLY AND BE BOUND BY THE TERMS OF THIS EULA. IF YOU DO NOT ACCEPT THE TERMS OF THIS EULA, YOU MAY NOT USE THE SOFTWARE. This EULA is a legal agreement between you and Sony Semiconductor Solutions Corporation (“SONY”). This EULA governs your rights and obligations regarding the SPRESENSE board system CPU firmware and GNSS firmware recovery tool software of SONY and/or its third party licensors (including SONY’s affiliates) and their respective affiliates (collectively, the “THIRD-PARTY SUPPLIERS”), together with any updates/upgrades provided by SONY, any printed, on-line or other electronic documentation for such software, and any data files created by operation of such software (collectively, the “SOFTWARE”). Notwithstanding the foregoing, any software in the SOFTWARE having a separate end user license agreement (including, but not limited to, GNU General Public license and Lesser/Library General Public License) shall be covered by such applicable separate end user license agreement in lieu of the terms of this EULA to the extent required by such separate end user license agreement (“EXCLUDED SOFTWARE”).

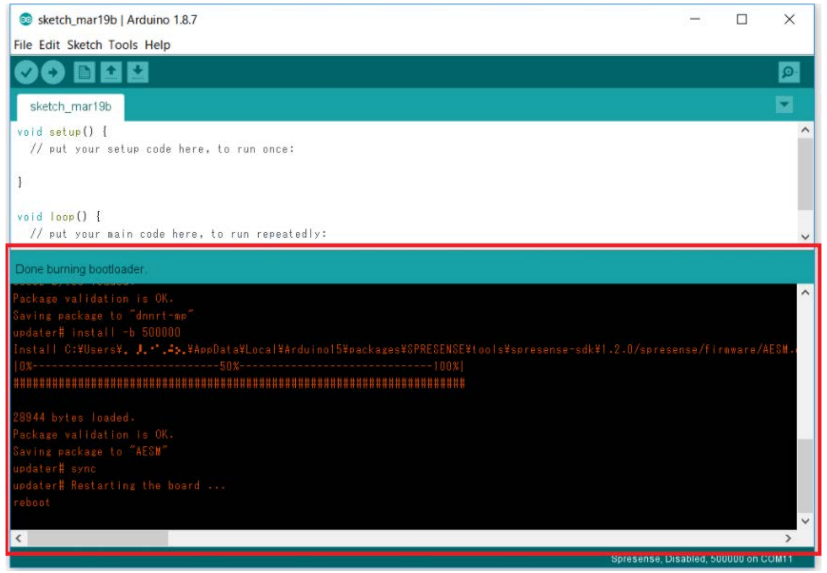
SOFTWARE LICENSE

The SOFTWARE is licensed, not sold. The SOFTWARE is protected by copyright and other intellectual property laws and international treaties.

COPYRIGHT

☑ I Accept the terms in the license agreement

STEP 6
Once the upload completes, this is how it should look like



sketch_mar19b | Arduino 1.8.7
File Edit Sketch Tools Help
sketch_mar19b
Spresense, Disabled, 500000 on COM11

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:
```

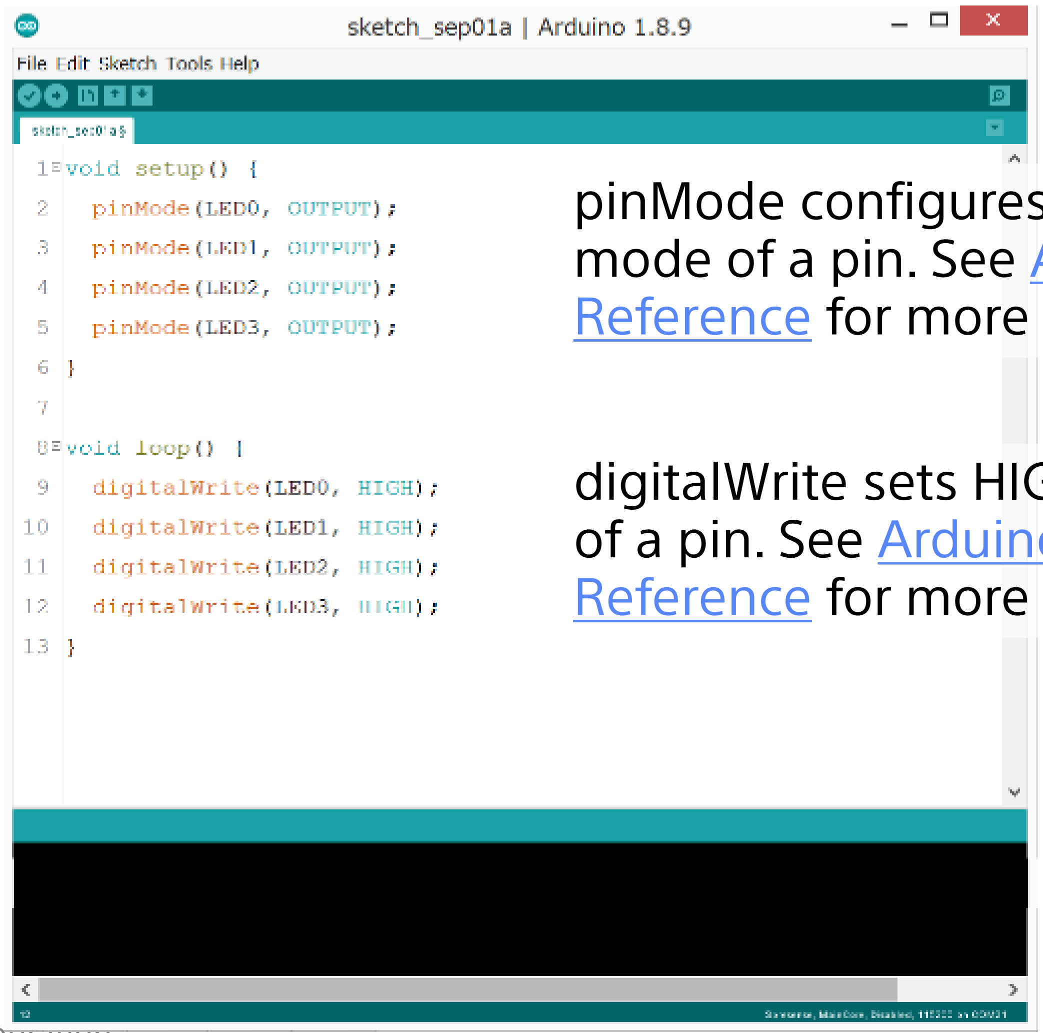
Done burning bootloader.

```
Package validation is OK.
Saving package to "dnnrt-mp"
updater# install -b 500000
Install C:¥Users¥[illegible]¥AppData¥Local¥Arduino15¥packages¥SPRESENSE¥tools¥spresense-sdk¥1.2.0/spresense/firmware/AESM.
|0%----------------------------50%----------------------------100%|
##################################################################

28944 bytes loaded.
Package validation is OK.
Saving package to "AESM"
updater# sync
updater# Restarting the board ...
reboot
```

Write your own "sketch" on Arduino IDE

setup function is called once just after system boot

loop function is called periodically during power on



```
void setup() {
  pinMode(LED0, OUTPUT);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
}

void loop() {
  digitalWrite(LED0, HIGH);
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, HIGH);
  digitalWrite(LED3, HIGH);
}
```

pinMode configures input/output mode of a pin. See Arduino Language Reference for more details

digitalWrite sets HIGH or LOW voltage of a pin. See Arduino Language Reference for more details

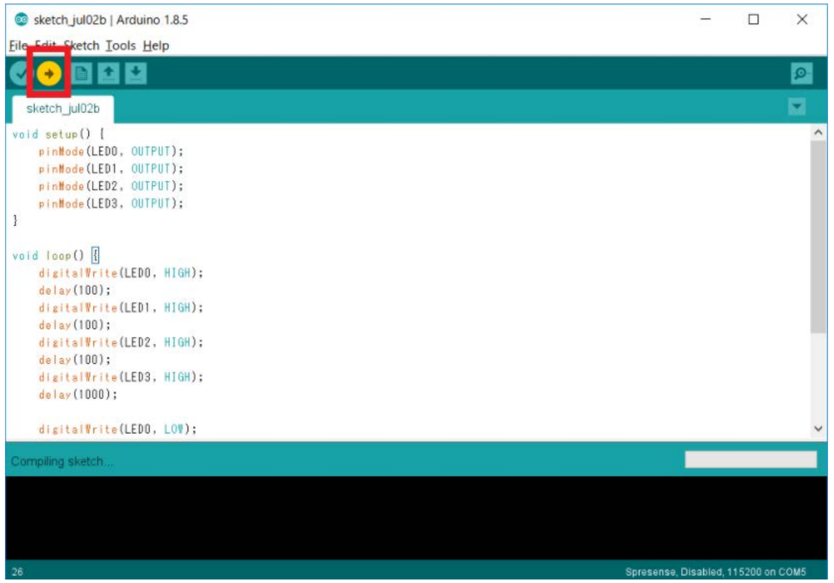
Write your own “sketch” on Arduino IDE

Copy and paste the following example LED test code into a new sketch:

```
void setup() {
    pinMode(LED0, OUTPUT);
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
}

void loop() {
    digitalWrite(LED0, HIGH);
    delay(100);
    digitalWrite(LED1, HIGH);
    delay(100);
    digitalWrite(LED2, HIGH);
    delay(100);
    digitalWrite(LED3, HIGH);
    delay(1000);

    digitalWrite(LED0, LOW);
    delay(100);
    digitalWrite(LED1, LOW);
    delay(100);
    digitalWrite(LED2, LOW);
    delay(100);
    digitalWrite(LED3, LOW);
    delay(1000);
}
```

Write your own “sketch” on Arduino IDE

STEP 1
Click update button

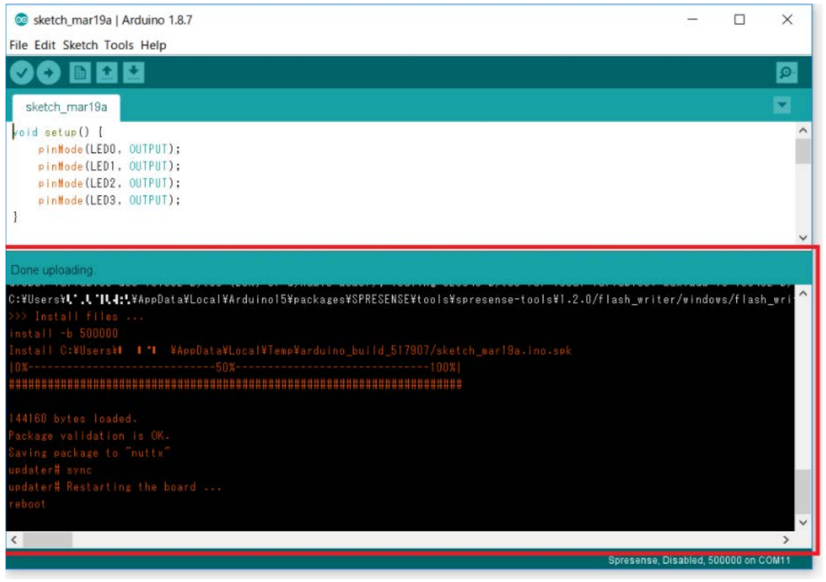


```
void setup() {
    pinMode(LED0, OUTPUT);
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
}

void loop() {
    digitalWrite(LED0, HIGH);
    delay(100);
    digitalWrite(LED1, HIGH);
    delay(100);
    digitalWrite(LED2, HIGH);
    delay(100);
    digitalWrite(LED3, HIGH);
    delay(1000);

    digitalWrite(LED0, LOW);
```

Compiling sketch...

26 Spresense, Disabled, 115200 on COM5

STEP 2
Wait for the upload to complete



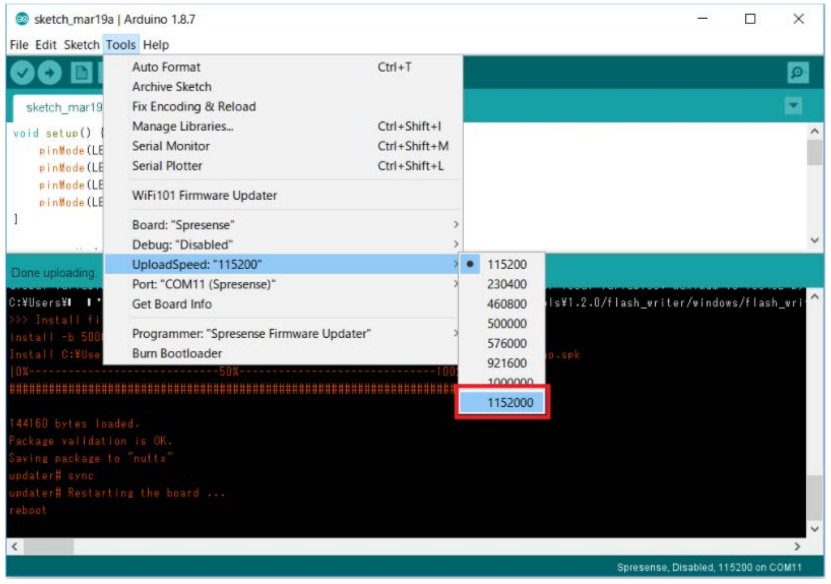
```
void setup() {
    pinMode(LED0, OUTPUT);
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
}
```

Done uploading.

```
C:¥Users¥[illegible]¥AppData¥Local¥Arduino15¥packages¥SPRESENSE¥tools¥spresense-tools¥1.2.0/flash_writer/windows/flash_wri
>>> Install files ...
install -b 500000
Install C:¥Users¥[illegible]¥AppData¥Local¥Temp¥arduino_build_517907/sketch_mar19a.ino.spk
|0%----------------------------50%----------------------------100%|
####################################################################

144160 bytes loaded.
Package validation is OK.
Saving package to "nuttx"
updater# sync
updater# Restarting the board ...
reboot
```

Spresense, Disabled, 500000 on COM11

STEP 3
Select baud rate



Done uploading.

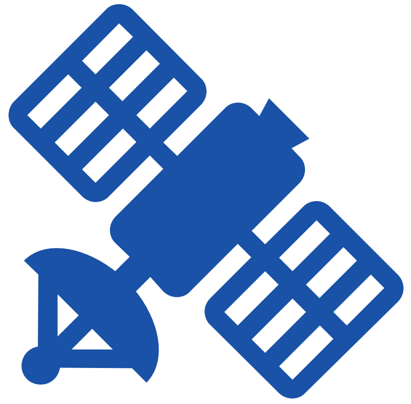
Spresense, Disabled, 115200 on COM11

※You may increase the baud rate to shorten the transfer/upload time. This may not always work depending on what hardware you use (PC, cables etc). A recommendation is to start at the default value and increase the baud rate to test if the transfer still works.

SONY

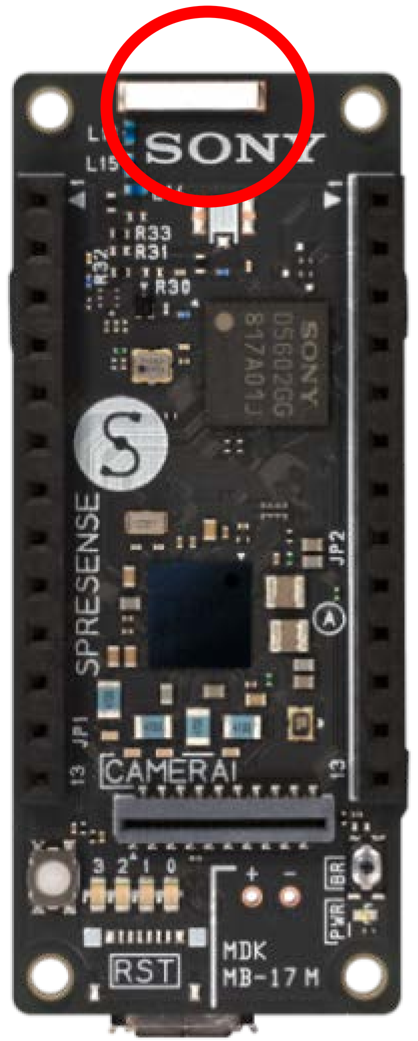
Tutorial 1

– How to use GNSS in SPRESENSE



GNSS (Global Navigation Satellite System)

GPS Chip Antenna

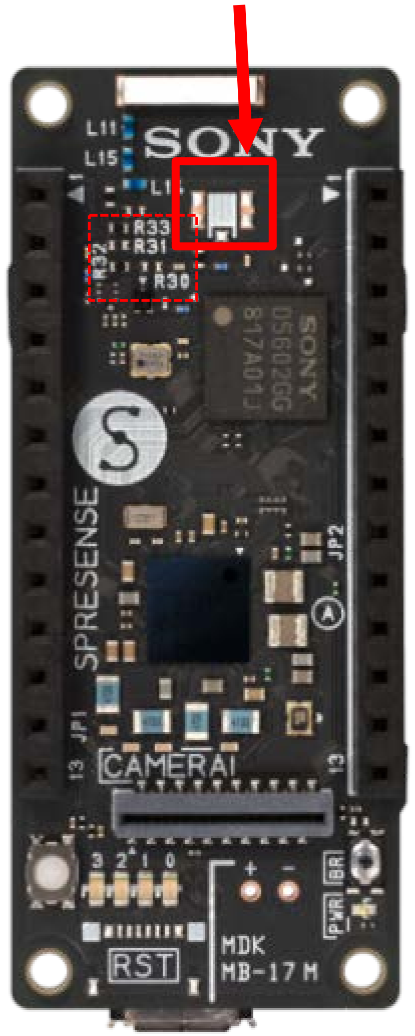


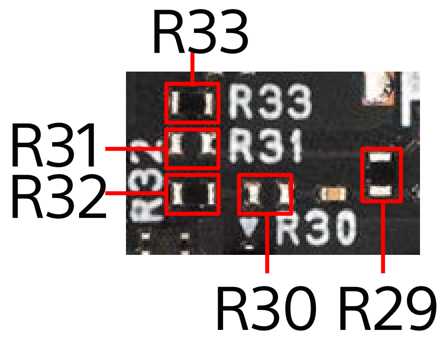
GPS (L1-CA)	GLONASS (L1)	QZSS (L1-C/A)	WAAS	QZSS (L1-S)	95% Accuracy ※	Effective place
○					< 5m	Under the open sky
○		○			< 5m	In East Asia and Oceania
○	○				< 7.8m	In the city
○	○	○			< 7.8m	In the city of East Asia and Oceania
○			○		< 2m	In North America
○		○		○	< 2m	In Japan

※ The positioning error(Accuracy) is a reference value under good conditions. It varies depending on your system and environment

Spresense with external GNSS antenna

Land for uFL connector



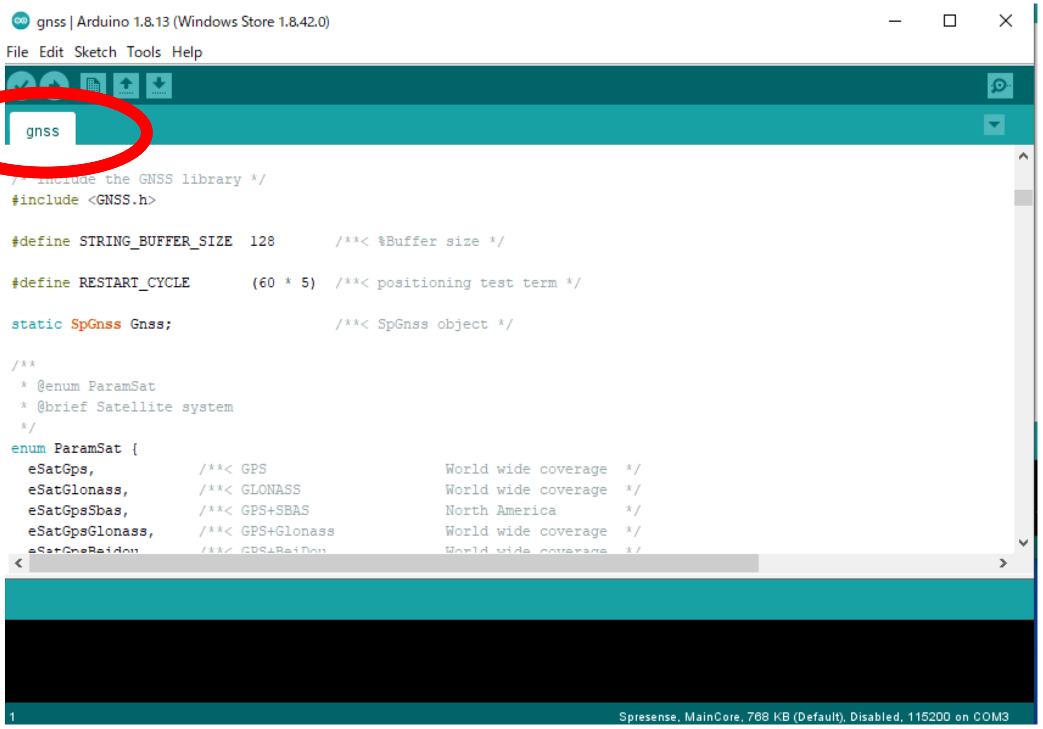


Antenna	R29	R31	R33	R30	R32	Note
Chip (on board)	CLOSE	OPEN	CLOSE	OPEN	CLOSE	default
Passive (external)	OPEN	CLOSE	OPEN	OPEN	CLOSE	
Active (external)	CLOSE	OPEN	CLOSE	CLOSE	OPEN	3.3V supplied to antenna

If you want to know further.
Please see the below site.
https://developer.sony.com/develop/spresense/docs/hw_docs_en.html#_how_to_use_the_external_antenna_for_gnss

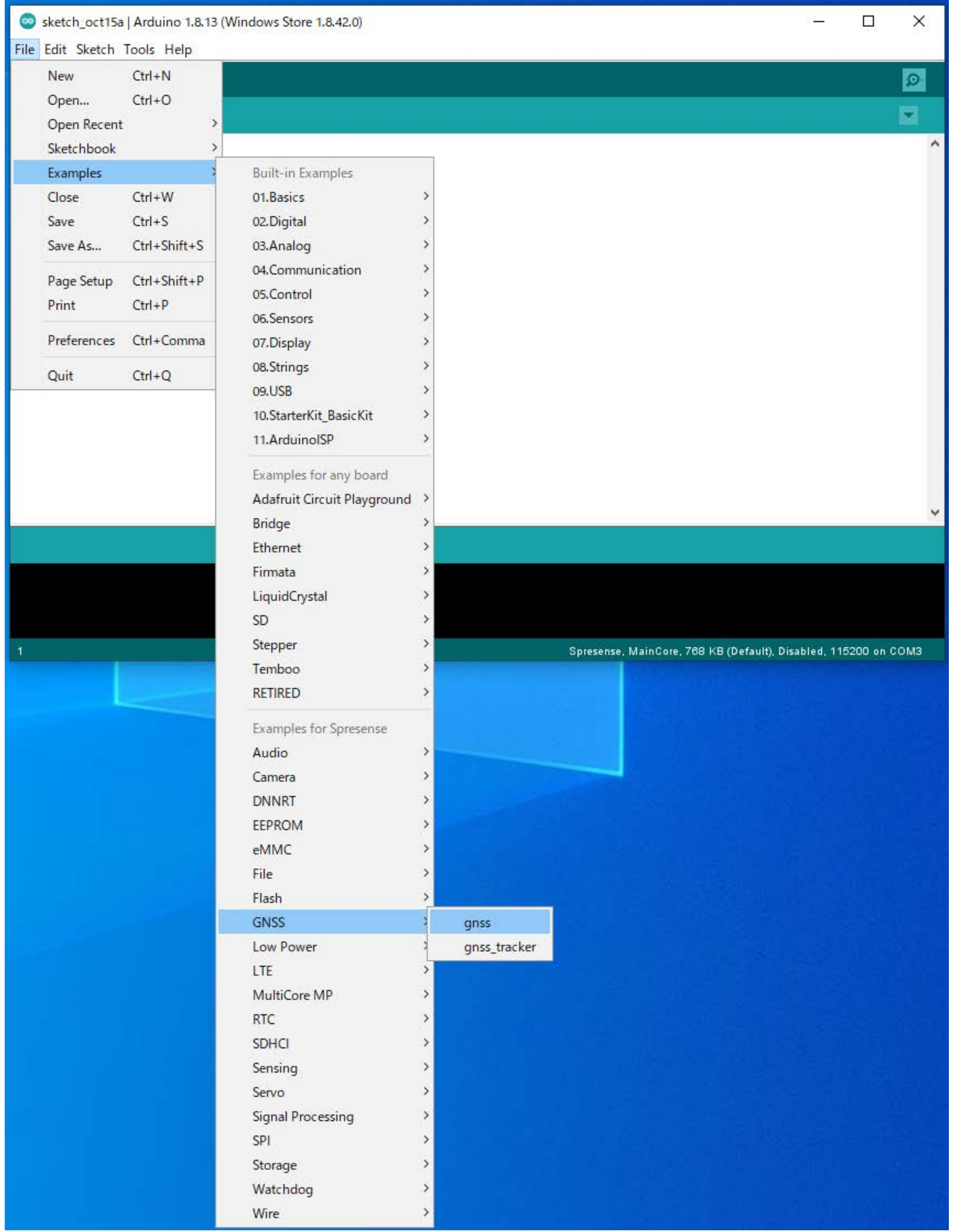
Open the GNSS sketch

- File → Examples → GNSS → gnss
- You can see the sketch in new window


gnss | Arduino 1.8.13 (Windows Store 1.8.42.0)
File Edit Sketch Tools Help
gnss

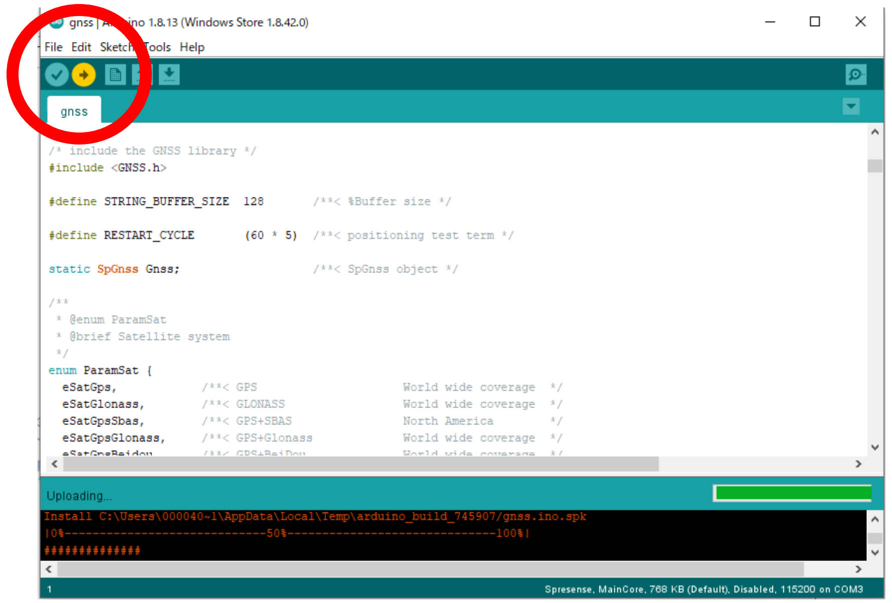
```
/* include the GNSS library */
#include <GNSS.h>

#define STRING_BUFFER_SIZE  128       /**< %Buffer size */

#define RESTART_CYCLE       (60 * 5)  /**< positioning test term */

static SpGnss Gnss;                   /**< SpGnss object */

/**
 * @enum ParamSat
 * @brief Satellite system
 */
enum ParamSat {
  eSatGps,            /**< GPS                     World wide coverage  */
  eSatGlonass,        /**< GLONASS                 World wide coverage  */
  eSatGpsSbas,        /**< GPS+SBAS                North America        */
  eSatGpsGlonass,     /**< GPS+Glonass             World wide coverage  */
```

```
1                         Spresense, MainCore, 768 KB (Default), Disabled, 115200 on COM3
```



Monitoring

- Upload

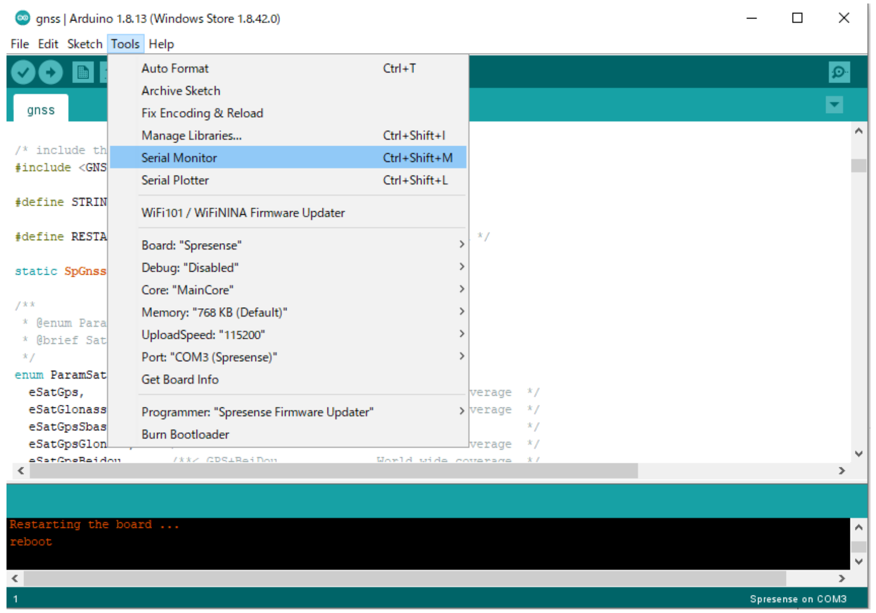

gnss | Arduino 1.8.13 (Windows Store 1.8.42.0)
File Edit Sketch Tools Help
gnss
Uploading...

```
/* include the GNSS library */
#include <GNSS.h>

#define STRING_BUFFER_SIZE  128       /**< %Buffer size */

#define RESTART_CYCLE       (60 * 5)  /**< positioning test term */

static SpGnss Gnss;                   /**< SpGnss object */

/**
 * @enum ParamSat
 * @brief Satellite system
 */
enum ParamSat {
  eSatGps,            /**< GPS                     World wide coverage  */
  eSatGlonass,        /**< GLONASS                 World wide coverage  */
  eSatGpsSbas,        /**< GPS+SBAS                North America        */
  eSatGpsGlonass,     /**< GPS+Glonass             World wide coverage  */
```

```
Install C:\Users\000040~1\AppData\Local\Temp\arduino_build_745907/gnss.ino.spk
|0%----------------------------50%----------------------------100%|
##############
```

Spresense, MainCore, 768 KB (Default), Disabled, 115200 on COM3

- Tool → Serial monitor


gnss | Arduino 1.8.13 (Windows Store 1.8.42.0)
File Edit Sketch Tools Help
gnss
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Get Board Info
Burn Bootloader

Board: "Spresense"
Debug: "Disabled"
Core: "MainCore"
Memory: "768 KB (Default)"
UploadSpeed: "115200"
Port: "COM3 (Spresense)"
Programmer: "Spresense Firmware Updater"

```
Restarting the board ...
reboot
```

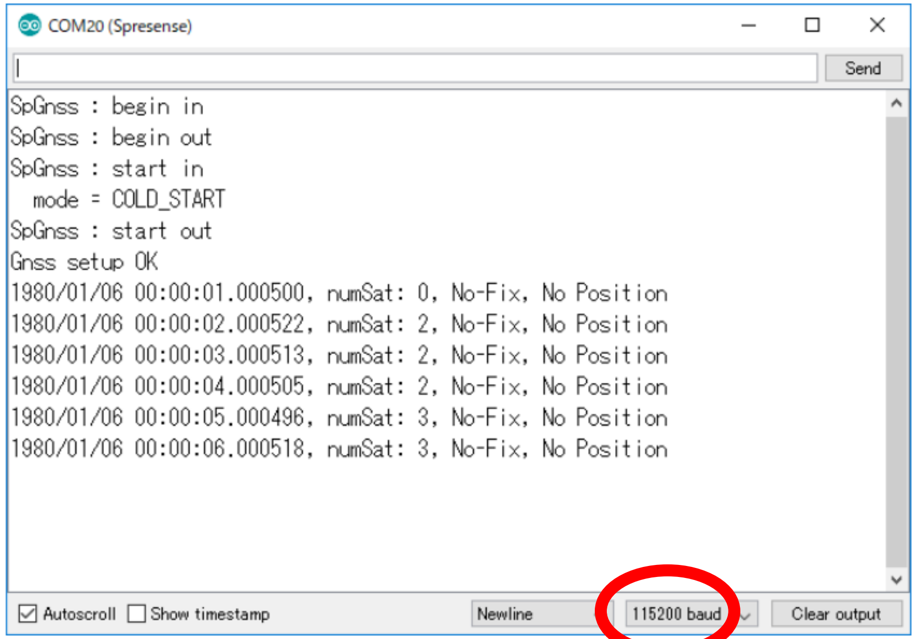
Spresense on COM3

Check the serial monitor

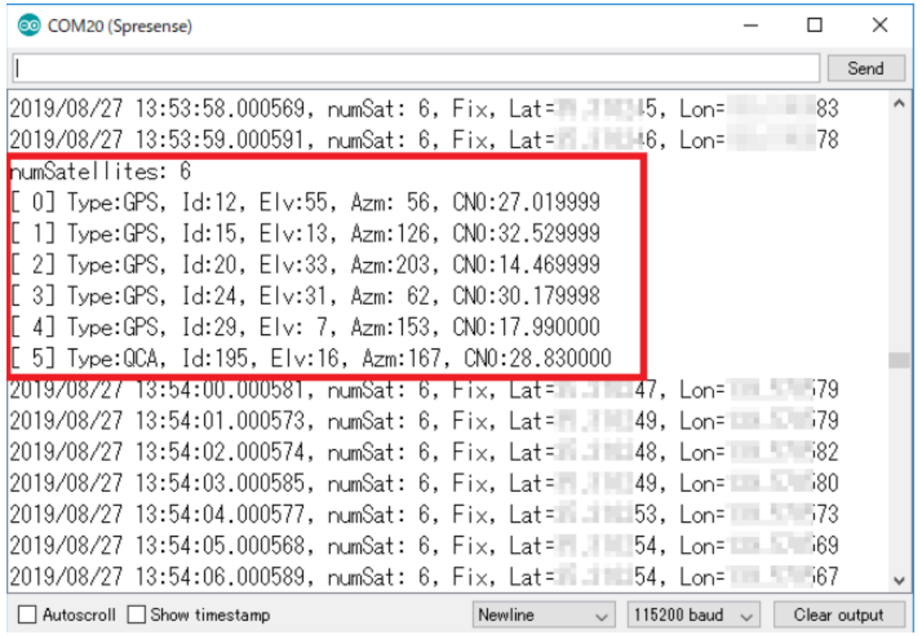
You can see the GNSS Info

- Before positioning is completed
  - Time stars from 1980/01/06, 00:00:00
  - numSat: Number of satellite
  - No-fix and No position are shown

- After positioning is completed
  - correct time, “Fix”, latitude and longitude
  - Type: types of satellite, id: satellite number
  - Elv: Elevation angle, Azm: Azimuth,
  - CN0: signal strength
    are shown



```
SpGnss : begin in
SpGnss : begin out
SpGnss : start in
  mode = COLD_START
SpGnss : start out
Gnss setup OK
1980/01/06 00:00:01.000500, numSat: 0, No-Fix, No Position
1980/01/06 00:00:02.000522, numSat: 2, No-Fix, No Position
1980/01/06 00:00:03.000513, numSat: 2, No-Fix, No Position
1980/01/06 00:00:04.000505, numSat: 2, No-Fix, No Position
1980/01/06 00:00:05.000496, numSat: 3, No-Fix, No Position
1980/01/06 00:00:06.000518, numSat: 3, No-Fix, No Position
```



```
2019/08/27 13:53:58.000569, numSat: 6, Fix, Lat=[redacted]5, Lon=[redacted]83
2019/08/27 13:53:59.000591, numSat: 6, Fix, Lat=[redacted]6, Lon=[redacted]78
numSatellites: 6
[ 0] Type:GPS, Id:12, Elv:55, Azm: 56, CN0:27.019999
[ 1] Type:GPS, Id:15, Elv:13, Azm:126, CN0:32.529999
[ 2] Type:GPS, Id:20, Elv:33, Azm:203, CN0:14.469999
[ 3] Type:GPS, Id:24, Elv:31, Azm: 62, CN0:30.179998
[ 4] Type:GPS, Id:29, Elv: 7, Azm:153, CN0:17.990000
[ 5] Type:QCA, Id:195, Elv:16, Azm:167, CN0:28.830000
2019/08/27 13:54:00.000581, numSat: 6, Fix, Lat=[redacted]47, Lon=[redacted]79
2019/08/27 13:54:01.000573, numSat: 6, Fix, Lat=[redacted]49, Lon=[redacted]79
2019/08/27 13:54:02.000574, numSat: 6, Fix, Lat=[redacted]48, Lon=[redacted]82
2019/08/27 13:54:03.000585, numSat: 6, Fix, Lat=[redacted]49, Lon=[redacted]80
2019/08/27 13:54:04.000577, numSat: 6, Fix, Lat=[redacted]53, Lon=[redacted]73
2019/08/27 13:54:05.000568, numSat: 6, Fix, Lat=[redacted]54, Lon=[redacted]69
2019/08/27 13:54:06.000589, numSat: 6, Fix, Lat=[redacted]54, Lon=[redacted]67
```

Overview of sketch

Declaration of header files, constants, objects, macros, functions

void setup() {

- Activate GNSS devices
- Select the satellite type
- Start positioning
- Declaration of methods printing time, position and number of satellite }

void loop() {

- Get and print the results of positioning (time, number of satellite, Id, latitude, longitude)
- stop and restart positioning every 5 minutes }

Sketch of GNSS (Global Positioning System)

```
#include <GNSS.h>
#define STRING_BUFFER_SIZE  128
#define RESTART_CYCLE (5*60)

static SpGnss Gnss;
```

Gnss is object of SpGnss class

```
enum ParamSat {
  eSatGps,            /**< GPS                    World wide coverage  */
  eSatGlonass,        /**< GLONASS                World wide coverage  */
  eSatGpsSbas,        /**< GPS+SBAS               North America        */
  eSatGpsGlonass,     /**< GPS+Glonass            World wide coverage  */
  eSatGpsBeidou,      /**< GPS+BeiDou             World wide coverage  */
  eSatGpsGalileo,     /**< GPS+Galileo            World wide coverage  */
  eSatGpsQz1c,        /**< GPS+QZSS_L1CA          East Asia & Oceania  */
  eSatGpsGlonassQz1c, /**< GPS+Glonass+QZSS_L1CA   East Asia & Oceania
*/
  eSatGpsBeidouQz1c,  /**< GPS+BeiDou+QZSS_L1CA    East Asia & Oceania
*/
  eSatGpsGalileoQz1c, /**< GPS+Galileo+QZSS_L1CA   East Asia & Oceania
*/
  eSatGpsQz1cQz1S,    /**< GPS+QZSS_L1CA+QZSS_L1S  Japan  */
};
```

Select satellite

```
/* Set this parameter depending on your current region. */
static enum ParamSat satType =  eSatGps;
```

```
static void Led_isActive(void) {
  static int state = 1;
  if (state == 1) {
    ledOn(PIN_LED0);
    state = 0;
  }
  else {
    ledOff(PIN_LED0);
    state = 1;
  }
}

static void Led_isPosfix(bool state) {
  if (state) {
    ledOn(PIN_LED1);
  }
  else {
    ledOff(PIN_LED1);
  }
}
static void Led_isError(bool state) {
  if (state) {
    ledOn(PIN_LED3);
  }
  else {
    ledOff(PIN_LED3);
  }
}
```

LED setting

Sketch of GNSS (Global Positioning System)

```
void setup() {
  int error_flag = 0;   /* put your setup code here, to run once: */

  Serial.begin(115200); /* Set serial baudrate. */

  sleep(3); /* Wait HW initialization done. */

  ledOn(PIN_LED0); /* Turn on all LED:Setup start. */
  ledOn(PIN_LED1);
  ledOn(PIN_LED2);
  ledOn(PIN_LED3);

  Gnss.setDebugMode(PrintInfo);

  int result;

  result = Gnss.begin();
```

Print debug messages about GNSS controlling and positioning if not set 0 to argument.

Activate GNSS device

```
 if (result != 0)  {
   Serial.println("Gnss begin error!!");
   error_flag = 1;
  }
  else {
   switch (satType) {
   case eSatGps:
     Gnss.select(GPS);
     break;

   case eSatGpsSbas:
     Gnss.select(GPS);
     Gnss.select(SBAS);
     break;

   case eSatGlonass:
     Gnss.select(GLONASS);
     break;

   case eSatGpsGlonass:
     Gnss.select(GPS);
     Gnss.select(GLONASS);
     break;

   case eSatGpsBeidou:
     Gnss.select(GPS);
     Gnss.select(BEIDOU);
     break;
```

$\Big\}$ the types of satellite

select(): Add specified satellite system to selection for positioning.

```
case eSatGpsGalileo:
  Gnss.select(GPS);
  Gnss.select(GALILEO);
  break;

 case eSatGpsQz1c:
  Gnss.select(GPS);
  Gnss.select(QZ_L1CA);
  break;

 case eSatGpsQz1cQz1S:
  Gnss.select(GPS);
  Gnss.select(QZ_L1CA);
  Gnss.select(QZ_L1S);
  break;

 case eSatGpsBeidouQz1c:
  Gnss.select(GPS);
  Gnss.select(BEIDOU);
  Gnss.select(QZ_L1CA);
  break;

 case eSatGpsGalileoQz1c:
  Gnss.select(GPS);
  Gnss.select(GALILEO);
  Gnss.select(QZ_L1CA);
  break;

 case eSatGpsGlonassQz1c:
 default:
  Gnss.select(GPS);
  Gnss.select(GLONASS);
  Gnss.select(QZ_L1CA);
  break;
}
```

the types of satellite

```
   result = Gnss.start(COLD_START);
   if (result != 0) {
    Serial.println("Gnss start error!!");
    error_flag = 1;
   }
   else {
    Serial.println("Gnss setup OK");
   }
  }

  ledOff(PIN_LED0); /* Turn off all LED:Setup done. */
  ledOff(PIN_LED1);
  ledOff(PIN_LED2);
  ledOff(PIN_LED3);

  /* Set error LED. */
  if (error_flag == 1) {
   Led_isError(true);
   exit(0);
  }
}

static void print_pos(SpNavData *pNavData) {
  char StringBuffer[STRING_BUFFER_SIZE];

  /* print time */
  snprintf(StringBuffer, STRING_BUFFER_SIZE, "%04d/%02d/%02d ", pNavData->time.year, pNavData->time.month, pNavData->time.day);
  Serial.print(StringBuffer);

  snprintf(StringBuffer, STRING_BUFFER_SIZE, "%02d:%02d:%02d.%06d, ", pNavData->time.hour, pNavData->time.minute, pNavData->time.sec, pNavData->time.usec);
  Serial.print(StringBuffer);
```

Start positioning

COLD_START: Discard all current position, time, and satellite orbital information. Start positioning from the beginning.

print_pos: print satellite positions

```
  /* print satellites count */
  snprintf(StringBuffer, STRING_BUFFER_SIZE, "numSat:%2d, ",
pNavData->numSatellites);
  Serial.print(StringBuffer);

  if (pNavData->posFixMode == FixInvalid) {
    Serial.print("No-Fix, "); /* print position data */
  }
  else {
    Serial.print("Fix, ");
  }
  if (pNavData->posDataExist == 0) {
    Serial.print("No Position");
  }
  else {
    Serial.print("Lat=");
    Serial.print(pNavData->latitude, 6);
    Serial.print(", Lon=");
    Serial.print(pNavData->longitude, 6);
  }

  Serial.println("");
}
```

numSatellites, posFixmode, posDataExist, latitude, longtitude are public attributes of SpNavData class

print_condition: print satellite condition

```
static void print_condition(SpNavData *pNavData) {
  char StringBuffer[STRING_BUFFER_SIZE];
  unsigned long cnt;

  snprintf(StringBuffer, STRING_BUFFER_SIZE, "numSatellites:%2d¥n", pNavData->numSatellites);
  Serial.print(StringBuffer); /* Print satellite count. */

  for (cnt = 0; cnt < pNavData->numSatellites; cnt++) {
    const char *pType = "---";
    SpSatelliteType sattype = pNavData->getSatelliteType(cnt);

    switch (sattype) {
      case GPS:
        pType = "GPS";
        break;

      case GLONASS:
        pType = "GLN";
        break;

      case QZ_L1CA:
        pType = "QCA";
        break;

      case SBAS:
        pType = "SBA";
        break;

      case QZ_L1S:
        pType = "Q1S";
        break;

      case BEIDOU:
        pType = "BDS";
        break;

      case GALILEO:
        pType = "GAL";
        break;

      default:
        pType = "UKN";
        break;
    }
```

Return types of all satellites which was received

all satellite types

```
 /* Get print conditions. */
   unsigned long Id  = pNavData->getSatelliteId(cnt);
   unsigned long Elv = pNavData->getSatelliteElevation(cnt);
   unsigned long Azm = pNavData->getSatelliteAzimuth(cnt);
   float sigLevel = pNavData->getSatelliteSignalLevel(cnt);

   /* Print satellite condition. */
   snprintf(StringBuffer, STRING_BUFFER_SIZE, "[%2d] Type:%s, Id:%2d,
Elv:%2d, Azm:%3d, CN0:", cnt, pType, Id, Elv, Azm );
   Serial.print(StringBuffer);
   Serial.println(sigLevel, 6);
  }
}

void loop() {
  static int LoopCount = 0;
  static int LastPrintMin = 0;

  Led_isActive(); /* Blink LED. */

  if (Gnss.waitUpdate(-1)) {
   SpNavData NavData;
   Gnss.getNavData(&NavData);

   /* Set posfix LED. */
   bool LedSet = (NavData.posDataExist && (NavData.posFixMode !=
FixInvalid));
   Led_isPosfix(LedSet);

   /* Print satellite information every minute. */
   if (NavData.time.minute != LastPrintMin) {
     print_condition(&NavData);
     LastPrintMin = NavData.time.minute;
   }
    /* Print position information. */
   print_pos(&NavData);
  }
```

Get the result of positioning

```
  else {
   /* Not update. */
   Serial.println("data not update");
  }

  LoopCount++; /* Check loop count. */
  if (LoopCount >= RESTART_CYCLE) {
   int error_flag = 0;
   ledOff(PIN_LED0); /* Turn off LED0 */
   Led_isPosfix(false); /* Set posfix LED. */

   /* Restart GNSS. */
   if (Gnss.stop() != 0){
    Serial.println("Gnss stop error!!");
    error_flag = 1;
   }
   else if (Gnss.end() != 0) {
    Serial.println("Gnss end error!!");
    error_flag = 1;
   }
   else {
    Serial.println("Gnss stop OK.");
   }

   if (Gnss.begin() != 0) {
    Serial.println("Gnss begin error!!");
    error_flag = 1;
   }
   else if (Gnss.start(HOT_START) != 0) {
    Serial.println("Gnss start error!!");
    error_flag = 1;
   }
   else {
    Serial.println("Gnss restart OK.");
   }
   LoopCount = 0;
   if (error_flag == 1) { /* Set error LED. */
    Led_isError(true);
    exit(0);
   }
  }
 }
}
```

Positioning stop and restart every 5 min
When you want to continue to get the position data, you can eliminate this loop process.

HOT_START: restart and get the position information immediately using the time, position and satellite condition which was obtained just before stop.
If you want to know further.
Please see the below site:
https://developer.sony.com/develop/spresense/docs/sdk_developer_guide_en.html#_start_positioning

GNSS Library API Reference

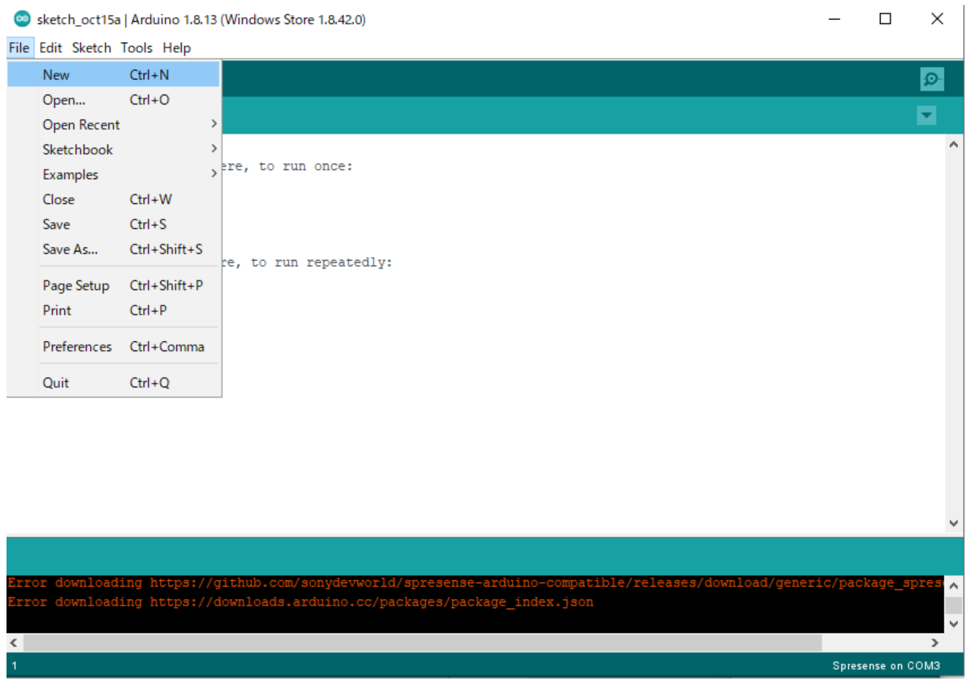
- SpGnss class
  - https://developer.sony.com/develop/spresense/developer-tools/api-reference/api-references-arduino/classSpGnss.html#ade8b8f5d5c05ba2ee67d857a32ef5229
- SpNavData class
  - https://developer.sony.com/develop/spresense/developer-tools/api-reference/api-references-arduino/classSpNavData.html#a29dd2b09d21ed4b7b60a19bce8de01b4

- You can get the detailed information of each APIs used in this tutorial.
- Red letters are API.

SONY

Tutorial 2
– How to get the disaster information from QZSS in SPRESENSE

- File → New



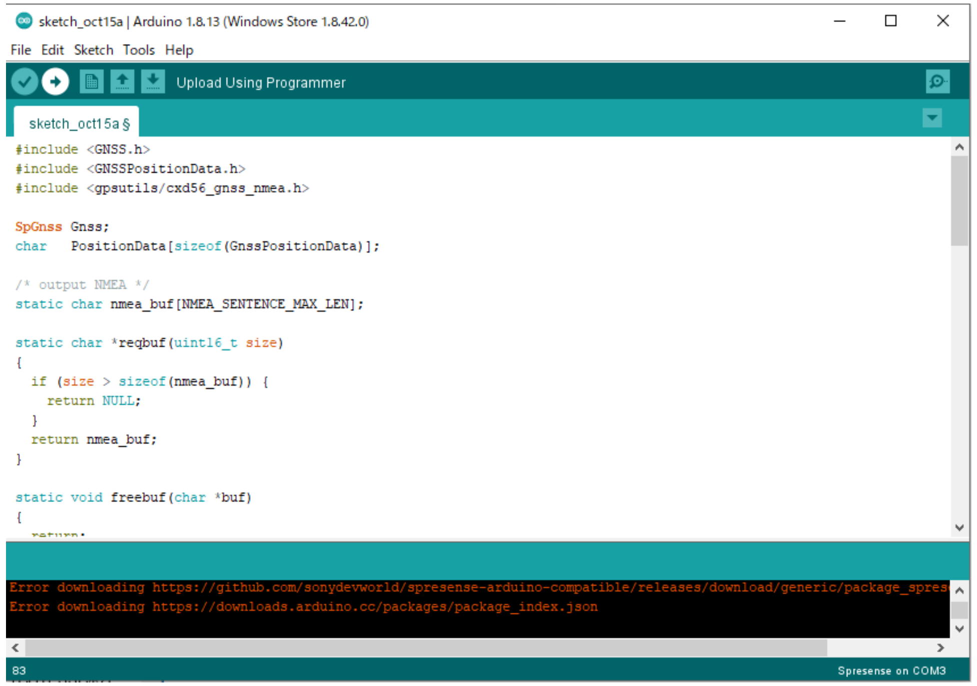
```
ere, to run once:

re, to run repeatedly:
```

```
Error downloading https://github.com/sonydevworld/spresense-arduino-compatible/releases/download/generic/package_spres
Error downloading https://downloads.arduino.cc/packages/package_index.json
```

- Copy and paste the source code in the following page and upload

or

- Open QZSS.ino in Arduino IDE and upload


sketch_oct15a | Arduino 1.8.13 (Windows Store 1.8.42.0)
File Edit Sketch Tools Help
Upload Using Programmer
sketch_oct15a §
83
Spresense on COM3

```
#include <GNSS.h>
#include <GNSSPositionData.h>
#include <gpsutils/cxd56_gnss_nmea.h>

SpGnss Gnss;
char   PositionData[sizeof(GnssPositionData)];

/* output NMEA */
static char nmea_buf[NMEA_SENTENCE_MAX_LEN];

static char *reqbuf(uint16_t size)
{
  if (size > sizeof(nmea_buf)) {
    return NULL;
  }
  return nmea_buf;
}

static void freebuf(char *buf)
{
  return;
```

```
Error downloading https://github.com/sonydevworld/spresense-arduino-compatible/releases/download/generic/package_spres
Error downloading https://downloads.arduino.cc/packages/package_index.json
```

Overview of sketch

Declaration of header files, constants, objects, macros, functions

void setup() {

- Activate GNSS devices
- Select the satellite type }

void loop() {

- Get GNSS position data
- Print QZQSM sentences }

Sketch of QZSS

```
#include <GNSS.h>
#include <GNSSPositionData.h>
#include <gpsutils/cxd56_gnss_nmea.h>

SpGnss Gnss;
char   PositionData[sizeof(GnssPositionData)];

/* output NMEA */
static char nmea_buf[NMEA_SENTENCE_MAX_LEN];

static char *reqbuf(uint16_t size) {
  if (size > sizeof(nmea_buf)) {
    return NULL;
  }
  return nmea_buf;
}

static void freebuf(char *buf) {
  return;
}

static int outbin(char *buf, uint32_t len) {
  return len;
}
```

Gnss is object of SpGnss class

Callback functions

```
static int outnmea(char *buf) {
  return printf("%s", buf);
}

void setup() {
  /* Initialize Serial */
  Serial.begin(115200);

  /* Initialize GNSS */
  if (Gnss.begin()) {
    Serial.println("begin error!");
  }

  /* select satellite system */
  Gnss.select(GPS);  //Gnss.select(GLONASS);
  Gnss.select(QZ_L1CA);
  Gnss.select(QZ_L1S);  //Gnss.select(SBAS);

  /* set interval */
  Gnss.setInterval(1);

  if (Gnss.start(COLD_START)) {
    Serial.println("start error!");
  }
```

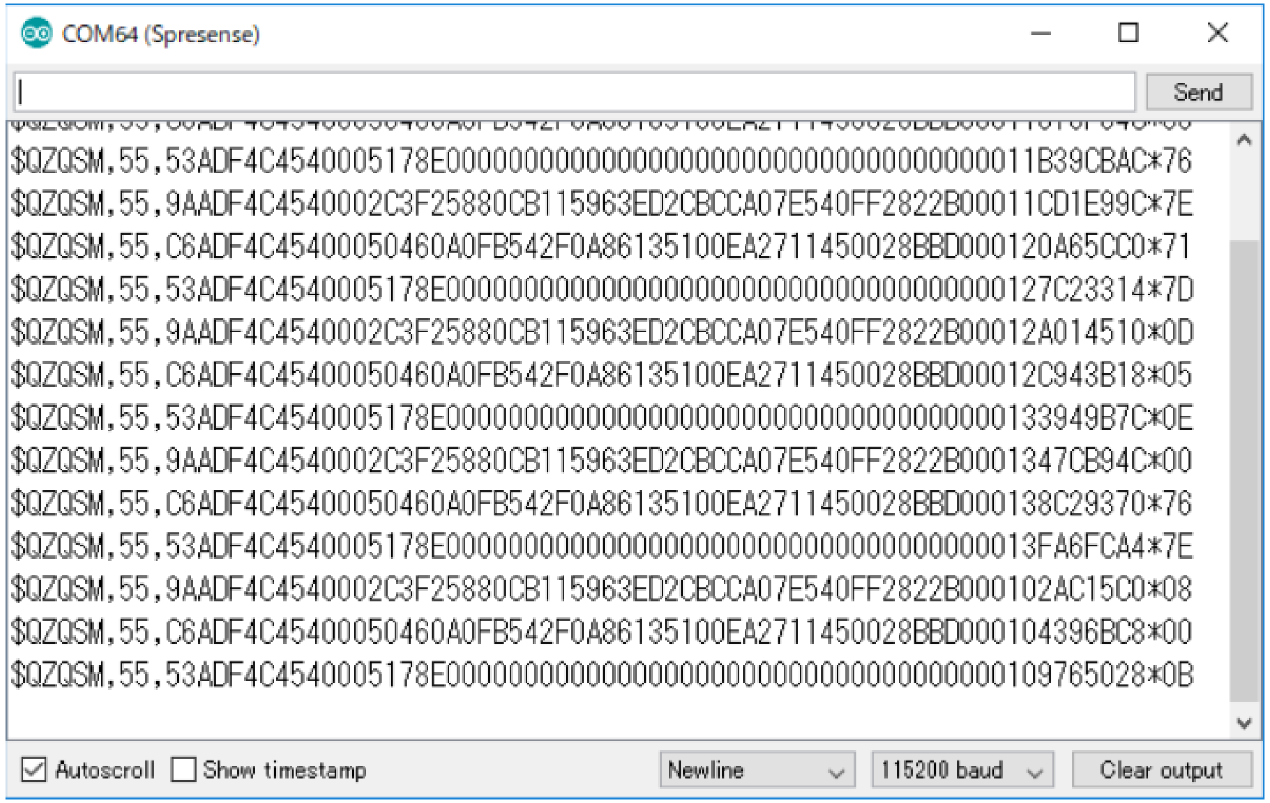
callback functions

Sketch of QZSS

```
  /* use NMEA library */
  NMEA_InitMask();
  NMEA_SetMask(0x4000); // only QZQSM
  NMEA_OUTPUT_CB  funcs;
  funcs.bufReq  = reqbuf;
  funcs.out     = outnmea;
  funcs.outBin  = outbin;
  funcs.bufFree = freebuf;
  NMEA_RegistOutputFunc(&funcs);
}

void loop()
{
  /* Check update. */
  if (Gnss.waitUpdate(1000)) {
    /* Output NMEA */
    Gnss.getPositionData(PositionData);
    NMEA_Output(&(((GnssPositionData*)PositionData)->Data));

    void *handle;
    if (handle = Gnss.getDCReport()) {
      NMEA_DcReport_Output(handle);
    }
  }
}
```

Get position data and print the QZQSM sentences

You can see the QZQSM sentences in serial monitor



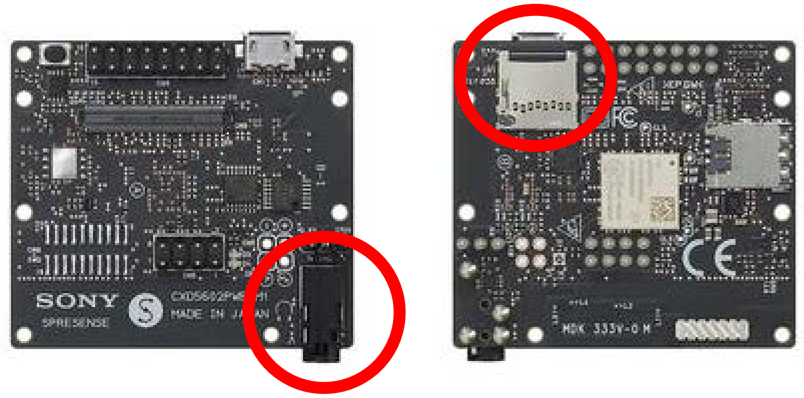
```
$QZQSM,55,53ADF4C4540005178E00000000000000000000000000000000000011B39CBAC*76
$QZQSM,55,9AADF4C4540002C3F25880CB115963ED2CBCCA07E540FF2822B00011CD1E99C*7E
$QZQSM,55,C6ADF4C45400050460A0FB542F0A86135100EA2711450028BBD000120A65CC0*71
$QZQSM,55,53ADF4C4540005178E000000000000000000000000000000000000127C23314*7D
$QZQSM,55,9AADF4C4540002C3F25880CB115963ED2CBCCA07E540FF2822B00012A014510*0D
$QZQSM,55,C6ADF4C45400050460A0FB542F0A86135100EA2711450028BBD00012C943B18*05
$QZQSM,55,53ADF4C4540005178E000000000000000000000000000000000000133949B7C*0E
$QZQSM,55,9AADF4C4540002C3F25880CB115963ED2CBCCA07E540FF2822B0001347CB94C*00
$QZQSM,55,C6ADF4C45400050460A0FB542F0A86135100EA2711450028BBD000138C29370*76
$QZQSM,55,53ADF4C4540005178E00000000000000000000000000000000000013FA6FCA4*7E
$QZQSM,55,9AADF4C4540002C3F25880CB115963ED2CBCCA07E540FF2822B000102AC15C0*08
$QZQSM,55,C6ADF4C45400050460A0FB542F0A86135100EA2711450028BBD000104396BC8*00
$QZQSM,55,53ADF4C4540005178E000000000000000000000000000000000000109765028*0B
```

SONY

Tutorial 3
– Play the recorded sound

Set up

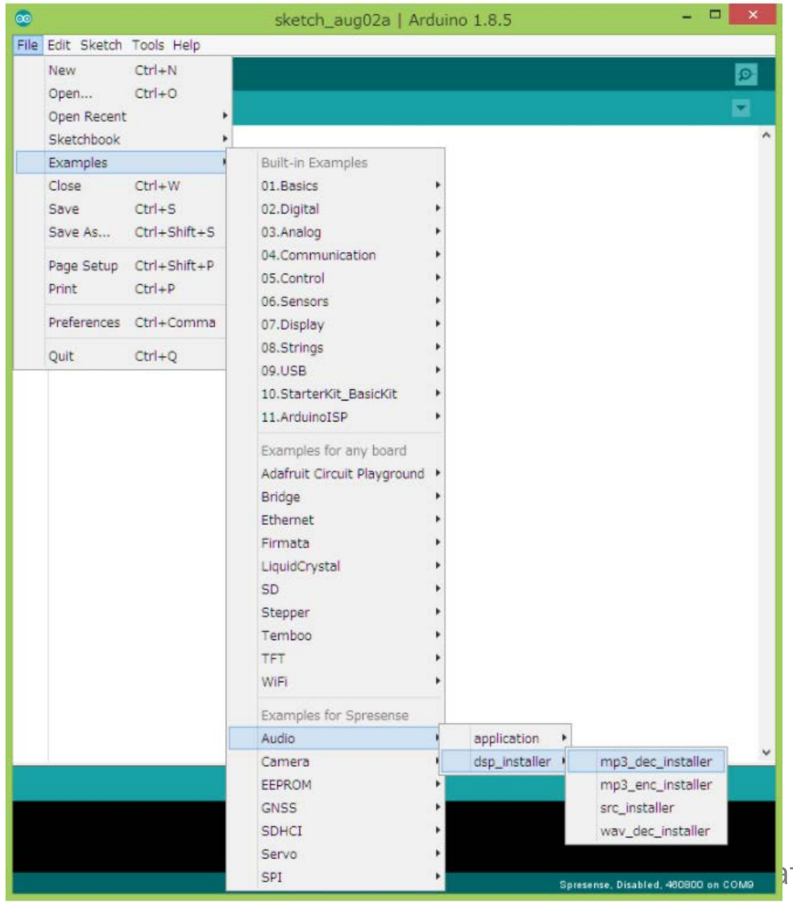
- We can use anything as recorder, smart phone, laptop and so on.
- Please recode the sound as MP3 file.
- Save the MP3 file in the micro SD card
- Insert SD card to LTE extension board
- Connect head-phone or speaker



Install MP3 Decoder to SPRESENSE

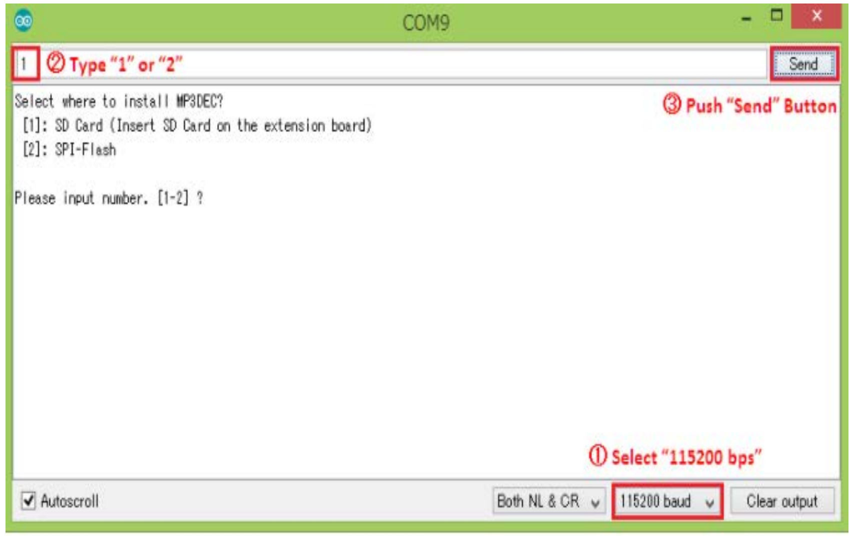
STEP 1

Select “mp3_dec_installer” and upload



STEP 2

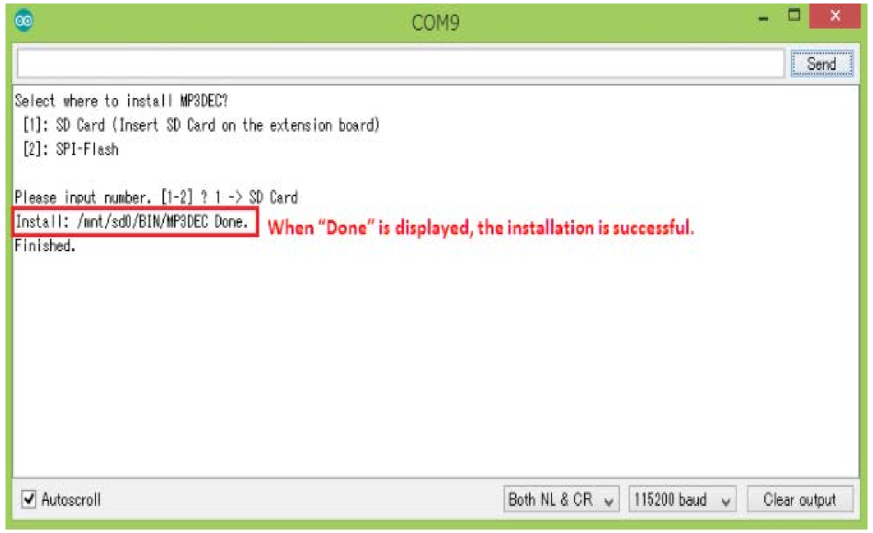
Select install type and baud rate



```
Select where to install MP3DEC?
 [1]: SD Card (Insert SD Card on the extension board)
 [2]: SPI-Flash

Please input number. [1-2] ?
```

STEP3

Check if the installation is done



```
Select where to install MP3DEC?
 [1]: SD Card (Insert SD Card on the extension board)
 [2]: SPI-Flash

Please input number. [1-2] ? 1 -> SD Card
Install: /mnt/sd0/BIN/MP3DEC Done.
Finished.
```

Overview of sketch

Declaration of header files, constants, objects, parameters, methods

void setup() {

- Initialize SD card
- Start Audio system
- Set clock mode, output device
- Initialize player with sampling rate, the location of mp3 file, stereo or monaural
- Read audio data
- Set the volume and start to play }

void loop() {

- Read audio data until file ends
- Stop player }

Sketch of code

```
#include <SDHCI.h>
#include <Audio.h>

SDClass theSD;
AudioClass *theAudio;

File myFile;
bool ErrEnd = false;

static void audio_attention_cb(const ErrorAttentionParam *atprm) {
  puts("Attention!");

  if (atprm->error_code >= AS_ATTENTION_CODE_WARNING) {
     ErrEnd = true;
   }
}

void setup() {
    while (!theSD.begin())  {
    Serial.println("Insert SD card."); /* wait until SD card is mounted. */
    }

  theAudio = AudioClass::getInstance(); // start audio system
  theAudio->begin(audio_attention_cb);

  puts("initialization Audio Library");

  theAudio->setRenderingClockMode(AS_CLKMODE_NORMAL);
  theAudio->setPlayerMode(AS_SETPLAYER_OUTPUTDEVICE_SPHP,
AS_SP_DRV_MODE_LINEOUT);
```

begin(): Initialize the audio library and HW modules

theAudio is instance of AudioClass

Set clock mode to normal

In case of I2S
AS_SETPLAYER_OUTPUTDEVICE_I2SOUTPUT

```
  err_t err = theAudio->initPlayer(AudioClass::Player0, AS_CODECTYPE_MP3,
"/mnt/sd0/BIN", AS_SAMPLINGRATE_AUTO, AS_CHANNEL_STEREO);

  /* Verify player initialize */
  if (err != AUDIOLIB_ECODE_OK) {
     printf("Player0 initialize error¥n");
     exit(1);
   }

  myFile = theSD.open("Sine.mp3"); /* Open file placed on SD card */

  /* Verify file open */
  if (!myFile) {
     printf("File open error¥n");
     exit(1);
   }
  printf("Open! %d¥n",myFile);

  /* Send first frames to be decoded */
  err = theAudio->writeFrames(AudioClass::Player0, myFile);

  if ((err != AUDIOLIB_ECODE_OK) && (err != AUDIOLIB_ECODE_FILEEND)){
     printf("File Read Error! =%d¥n",err);
     myFile.close();
     exit(1);
   }

  puts("Play!");

  theAudio->setVolume(30); /* Main volume [dB] set */
  theAudio->startPlayer(AudioClass::Player0);
}
```

Initialize player

location of mp3 decoder

In case of monaural
AS_CHANNEL_MONO

Only mp3 allows to set AUTO

Read audio data from SD card and write it to FIFO

Start to play

Sketch of code

```
void loop() {
  puts("loop!!");

  /* Send new frames to decode in a loop until file ends */
  int err = theAudio->writeFrames(AudioClass::Player0, myFile);

  /*  Tell when player file ends */
  if (err == AUDIOLIB_ECODE_FILEEND) {
      printf("Main player File End!¥n");
    }

  /* Show error code from player and stop */
  if (err) {
      printf("Main player error code: %d¥n", err);
      goto stop_player;
    }

  if (ErrEnd) {
      printf("Error End¥n");
      goto stop_player;
    }

  /* This sleep is adjusted by the time to read the audio stream file.
     Please adjust in according with the processing contents
     being processed at the same time by Application.
  */
  usleep(40000);
```

Read audio data from SD card and write it to FIFO

```
  /* Don't go further and continue play */
  return;

stop_player:
  sleep(1);
  theAudio->stopPlayer(AudioClass::Player0);
  myFile.close();
  exit(1);
}
```

Stop Audio

Audio Library API Reference

- https://developer.sony.com/develop/spresense/developer-tools/api-reference/api-references-arduino/classAudioClass.html#ae740705f6a3e40a94546fc4e70950925
- You can get the detailed information of each APIs used in this tutorial.
- Red letters are API

SONY