SONY

Online lecture and demonstration of Spresense

R&D Center Tokyo Laboratory 14
Kentaro Matsuura
Satoru Iwasaki

Agenda

- Review of Homework Video (https://www.youtube.com/watch?v=GYBl8eo86uA)
  - What is Spresense ?
  - Edge AI with Spresense

- Demonstration
  - Overview, the alert for the specific disaster by QZSS and SPRESENSE
  - Short review of tutorial
    - How to use GNSS in SPRESENSE
    - How to get the disaster information from QZSS in SPRESENSE
    - Play the recoded sound
  - Detailed explanation of the demonstration

SONY

Review of Homework Video

What is SPRESENSE ?

Low power Multi Processor

SPRESENSE has a brand new low power multi-core processor that has 6 ARM® Cortex® M4F being the capability of max frequency at 156MHz and driven at 0.7V

High Resolution Audio Output

Not only has the capability of playing High-Resolution Audio of 192kHz/24bit but also has a built-in D-class amplifier enabling BLT-stereo output.

Multiple Micro-phone inputs

You can enjoy 4 mic inputs with an analog microphone or up to 8 mic inputs with a digital microphone. All of the channels can record at 192kHz/24bit simultaneously.

Camera interface

Support Sony's 5M pixels CMOS (Exmore) sensor with dedicated CMOS 8 parallel interface.

Built-in GNSS (GPS)

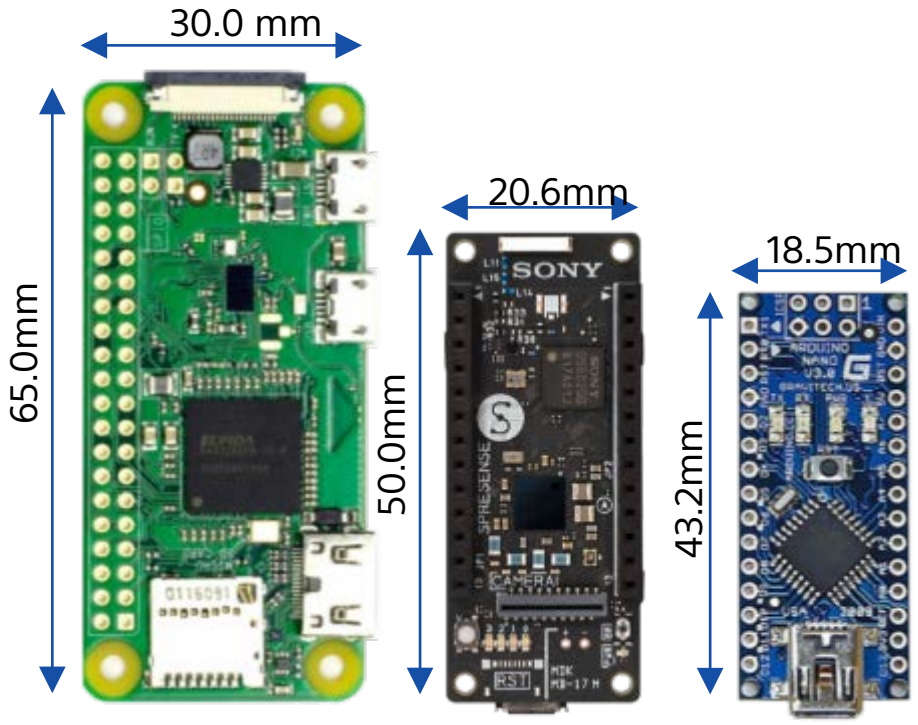
Built-in GNSS function supporting GPS/GLONASS/QZSS allows you to get a precise position all over the world

Artificial Intelligence

SPRESENSE can accommodate AI functionality made by Sony Neural Network Console

LOW POWER, but HIGH PERFORMANCE

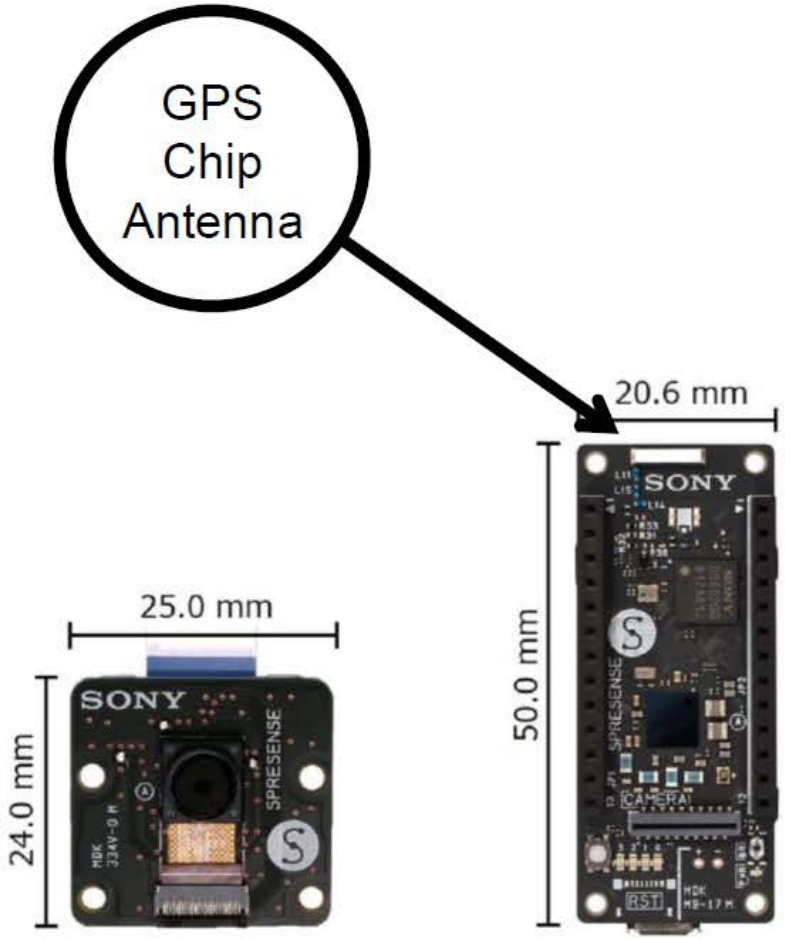
Performance comparison



	Raspberry Pi Zero WH	SPRESENSE™ Main board	Arduino Nano 3.0
Power Consumption*1	500mW*2	30mW	100mW
Calculation Power	1250DMIPS (ARM11 Single Core)	1170DMIPS (Cortex M4F Six-Cores)	20DMIPS (ATmega328p)
Board size	65.0mm 30.0mm	50.0mm 20.6mm	43.2mm 18.5mm
Others	Wi-Fi/Bluetooth Display output Audio output Camera interface	GNSS receiver Hi Reso. audio input and output Camera interface	

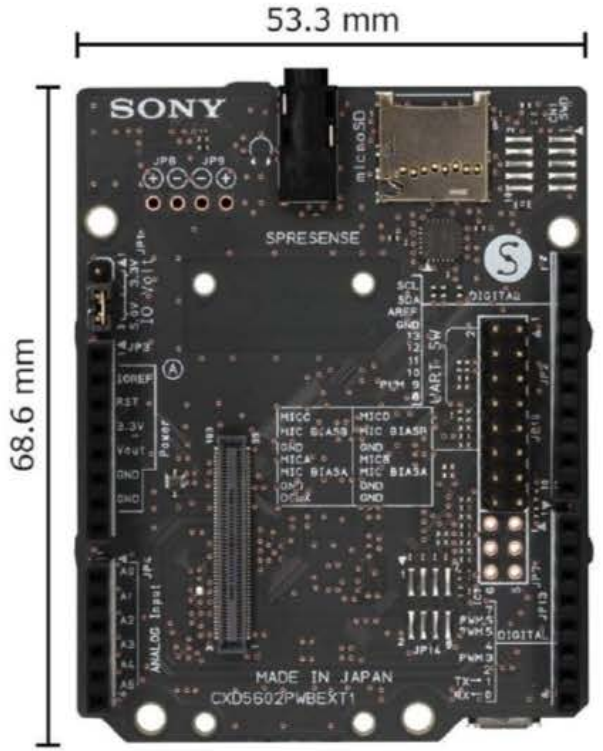
*1 No load *2 Wi-Fi/Bluetooth are OFF

SPRESENSE Peripheral boards

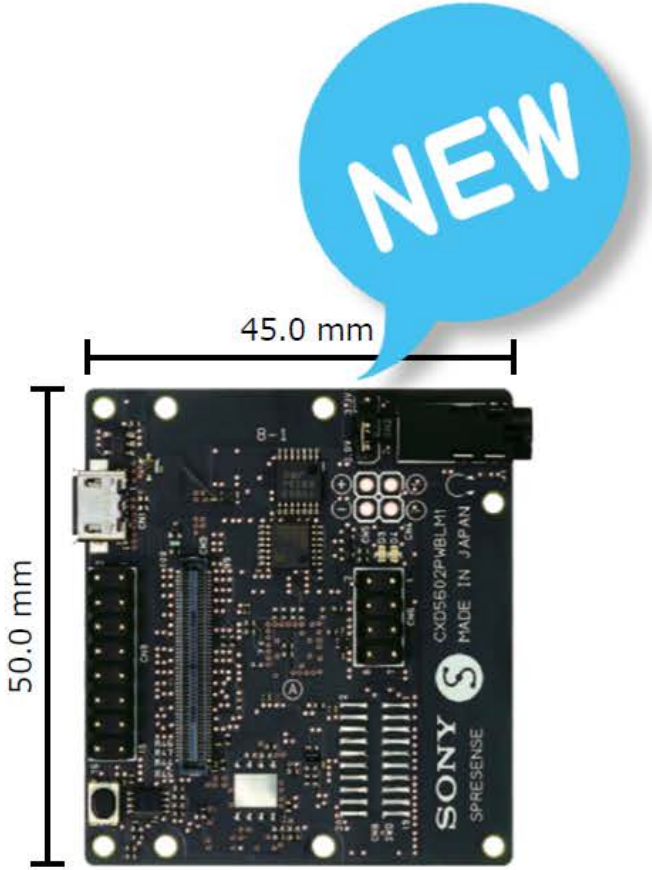


CAMERA BOARD

MAIN BOARD



EXTENSION BOARD



LTE-M EXTENSION BOARD

SPRESENSE Peripheral boards

Wi-Fi Add-on

BLE Add-on

Sigfox Add-on

LoRa-BLE Add-on

Sensor Add-on

Sensor Add-on

Sensor Add-on

Flexible and Expandable

Sandwich Concept Design

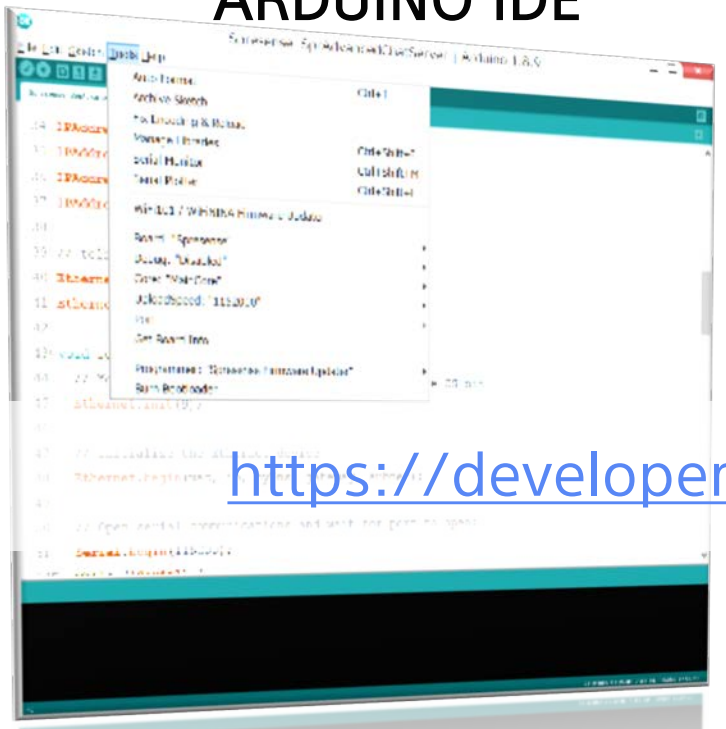
Portable Player Extension Board

Extension Board

LTE(Cat-M) Extension Board

SPRESENSE Development Environment

ARDUINO IDE



VSCode

https://developer.sony.com/ja/develop/spresense/

SCRATCH 3rd party

Already available
https://www.neusoft.co.jp

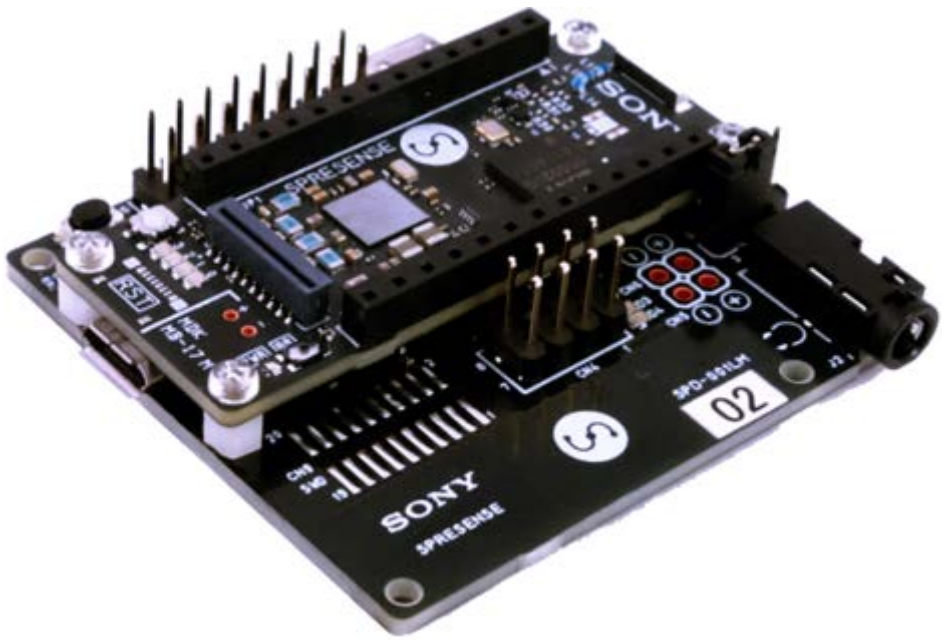
python™ 3rd party

Already available
https://www.zerynth.com/zerynth-studio/

Java MICROEJ® 3rd party

Already available
https://www.microej.com/product/studio/

SPRESENSE LTE(Cat-M) Extension Board

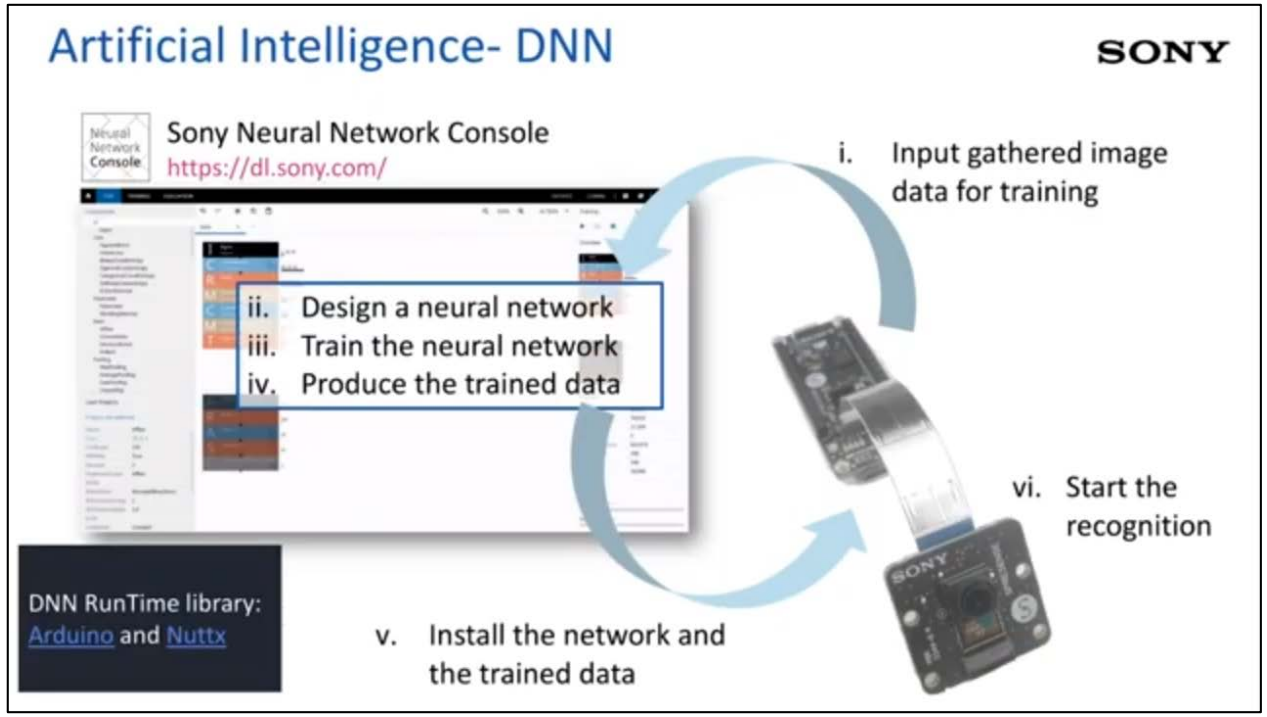


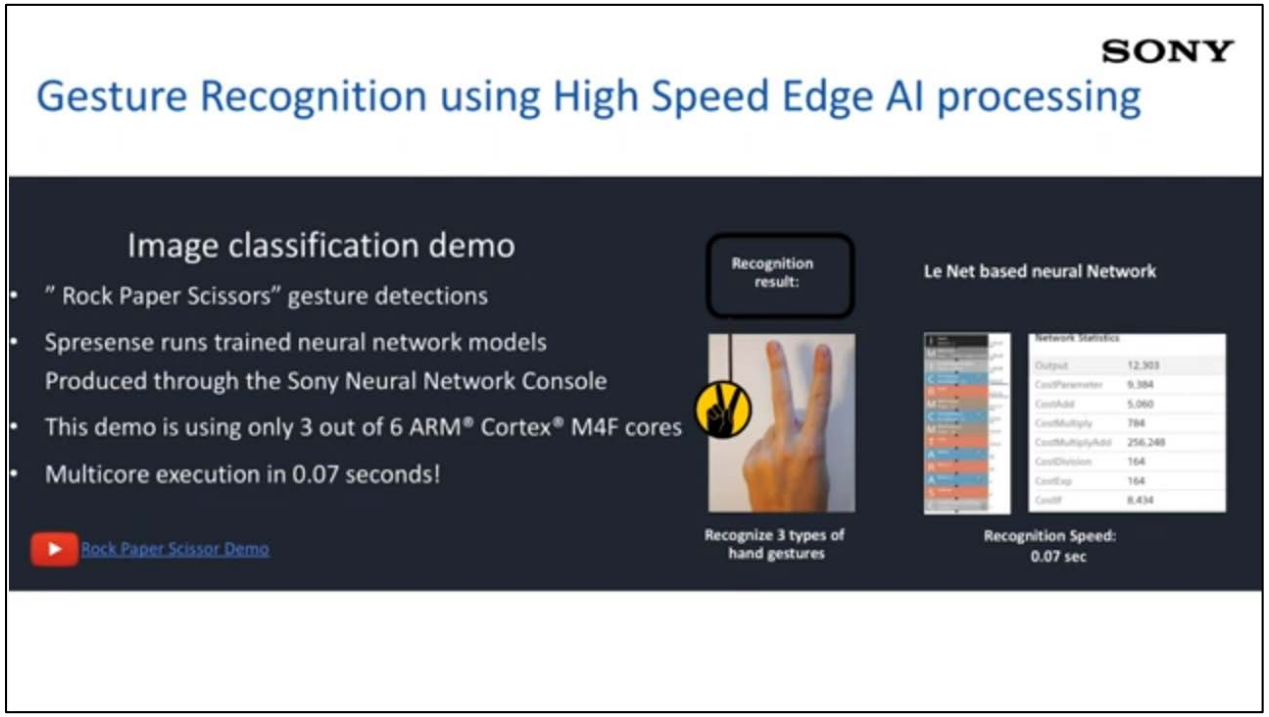
SPRESENSE LTE(Cat-M) Extension Board		
Size	50mmx45mmx1.6mm（TBD）	
Module name	LBAD0XX1SC（ALT1250）	
Communication System	LTE Cat-M1	
Band	Band 1, Band 3, Band 8, Band 18, Band 19, Band 41	
Audio input / output Digital input / output	Analog MIC x 2 or Digital MIC x 4 HeadPhone Stereo (Line-out)	
	SPI x 1（Master） PWM x 4 GPIO x 4	3.3V or 5.0V (selectable)
Analog input	Analog Input x 2 (5.0V range)	
External memory interface	SD Card Interface	
Antenna	On board	
SIM	SIM card	

SONY

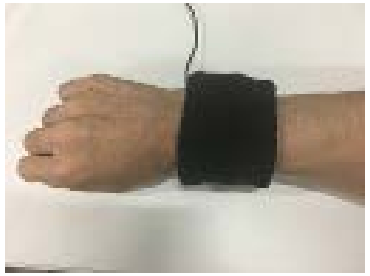
Edge AI with Spresense

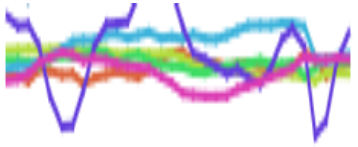
Edge AI with Spresense



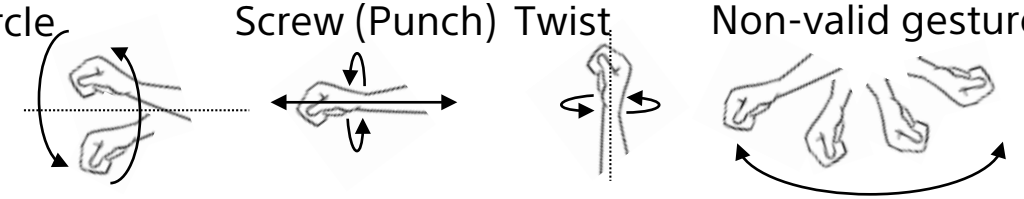


Other applications



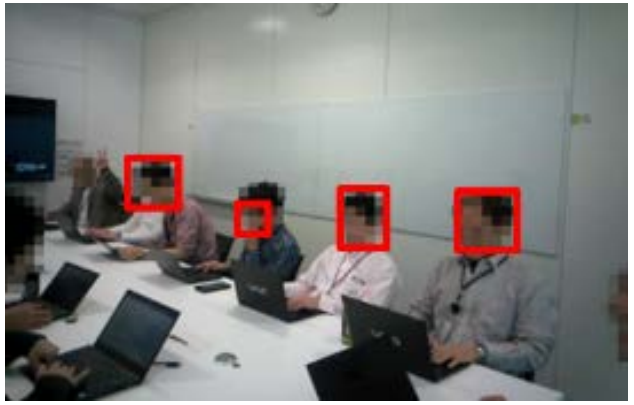


Accel + Gyro
6 axis data



No-touch operation by hand motion recognition

Count number of humans from heatmap



Detect the position of each person

What is SPRESENSE ?

Low power Multi Processor

SPRESENSE has a brand new low power multi-core processor that has 6 ARM® Cortex® M4F being the capability of max frequency at 156MHz and driven at 0.7V

High Resolution Audio Output

Not only has the capability of playing High-Resolution Audio of 192kHz/24bit but also has a built-in D-class amplifier enabling BLT-stereo output.

Multiple Micro-phone inputs

You can enjoy 4 mic inputs with an analog microphone or up to 8 mic inputs with a digital microphone. All of the channels can record at 192kHz/24bit simultaneously.

Camera interface

Support Sony's 5M pixels CMOS (Exmore) sensor with dedicated CMOS 8 parallel interface.

Built-in GNSS (GPS)

Built-in GNSS function supporting GPS/GLONASS/QZSS allows you to get a precise position all over the world

Artificial Intelligence

SPRESENSE can accommodate AI functionality made by Sony Neural Network Console

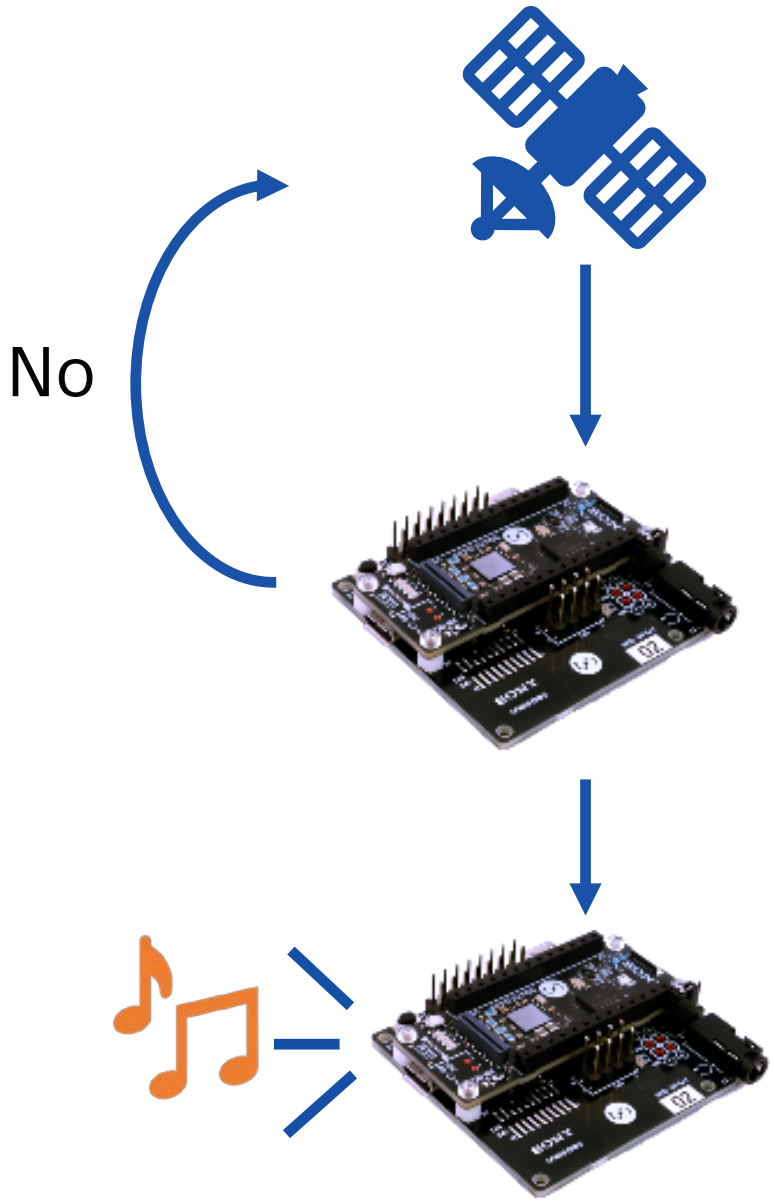
- In this program, you will make protypes using Spresense, especially for GNSS and QZSS.
- We would be glad if you keep in mind that SPRESENSE has **many other possibility.**

SONY

Demonstration

- Overview, the Alert for the specific disaster by QZSS and SPRESENSE

Overview



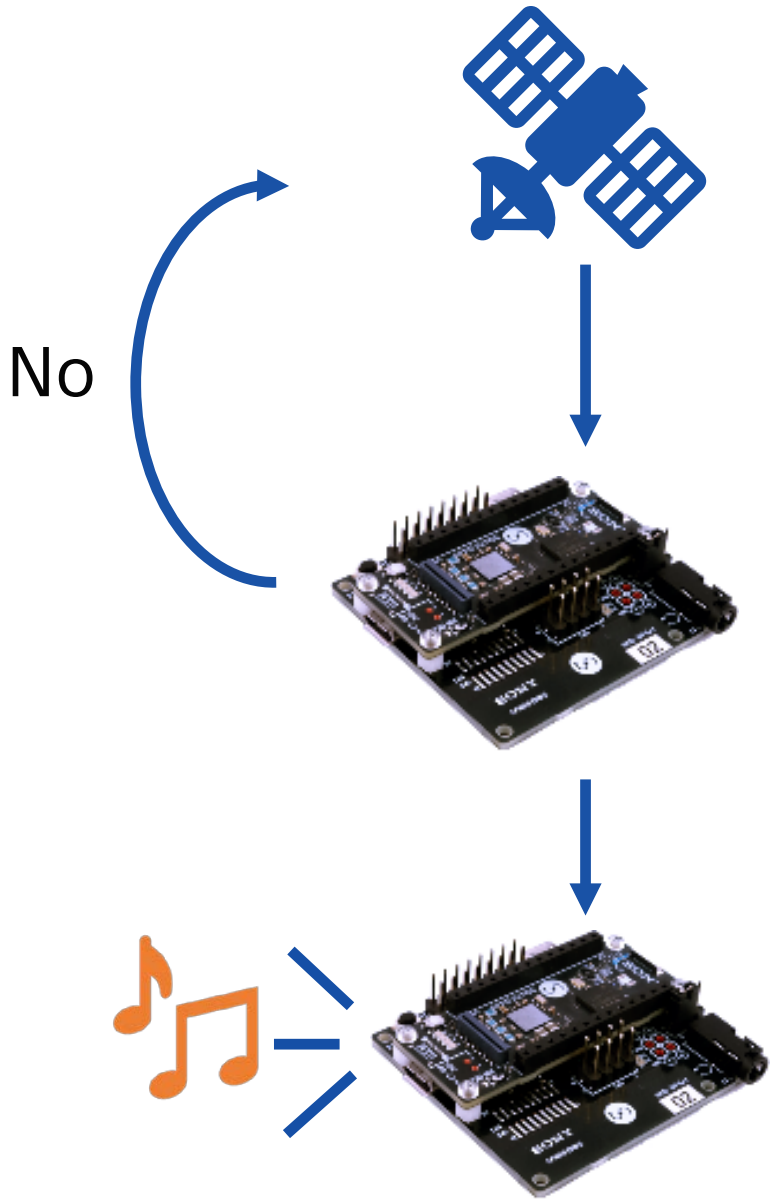
Reception of the disaster info as QZQSM sentence

Decode of all QZQSM sentences and check if the disaster info which you want was received

Yes

Play the recoded sound

Overview



Tutorial 1, 2

Reception of the disaster info as QZQSM sentence

Decode of all QZQSM sentences and check if the disaster info which you want was received

Yes

Tutorial 3

Play the recoded sound

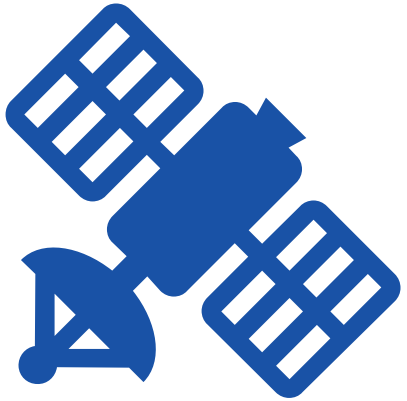
SONY

Q & A

SONY

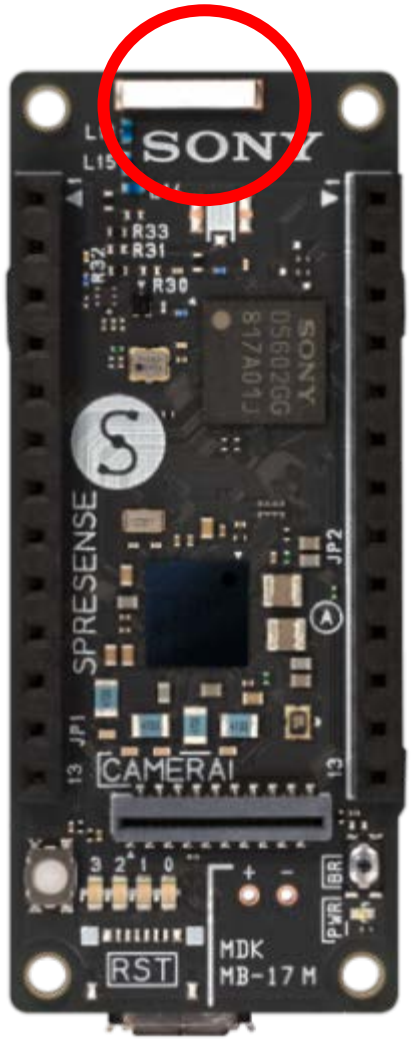
Tutorial 1

– How to use GNSS in SPRESENSE



GNSS (Global Navigation Satellite System)

GPS Chip Antenna

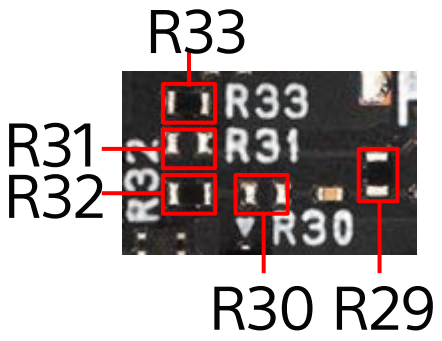


GPS (L1-CA)	GLONASS (L1)	QZSS (L1-C/A)	WAAS	QZSS (L1-S)	95% Accuracy ※	Effective place
○					< 5m	Under the open sky
○		○			< 5m	In East Asia and Oceania
○	○				< 7.8m	In the city
○	○	○			< 7.8m	In the city of East Asia and Oceania
○			○		< 2m	In North America
○		○		○	< 2m	In Japan

※ 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



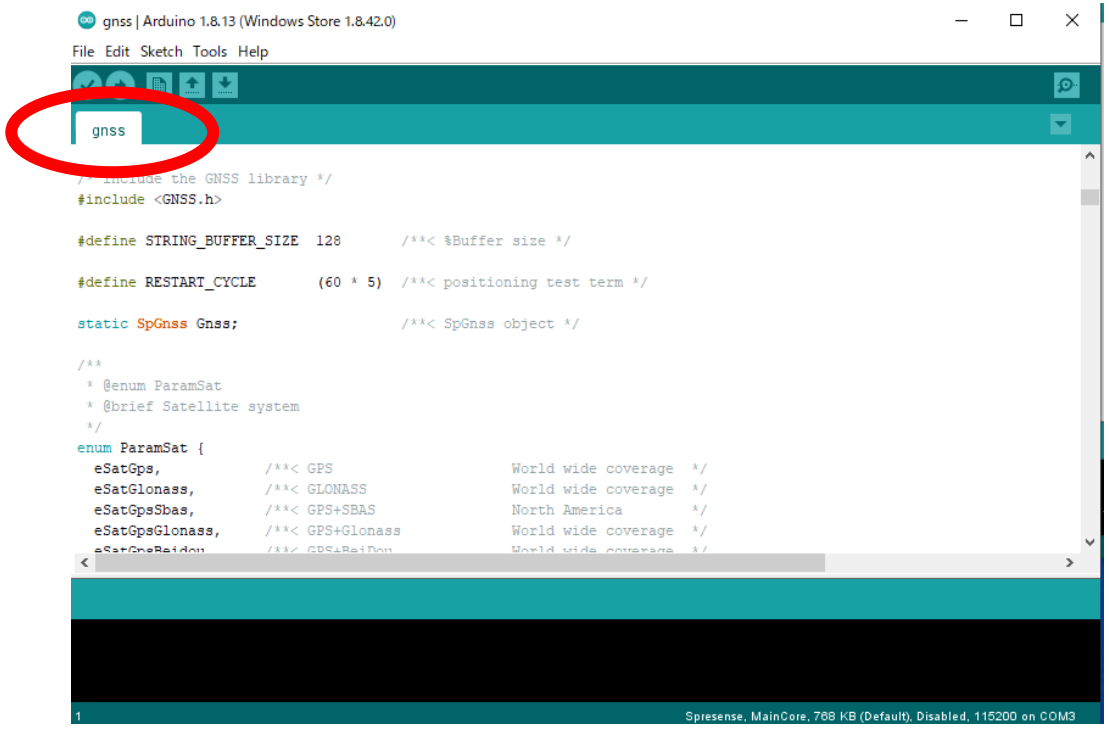
Antenna	R29	R31	R33	R30	R32	Note
Chip (on board)	CLOSE	OPEN	CLOSE	OPEN	CLOSE	default
Passive (external)	OPEN	CLOSE	OPEN	OPEN	CLOSE	
Active (external)	CLOSE	OPEN	CLOSE	CLOSE	OPEN	3.3V supplied to antenna

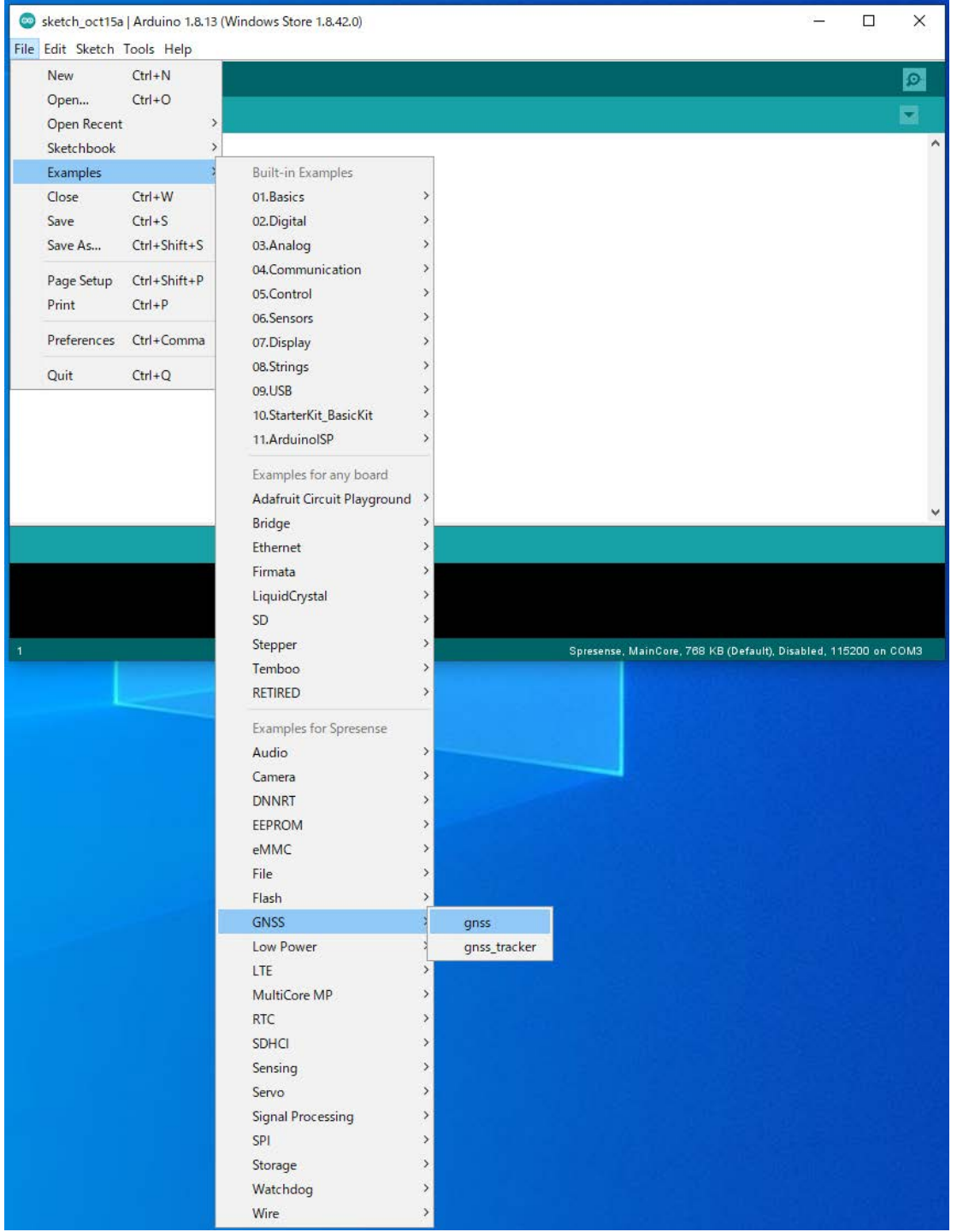
If you want to know further.
Please see the below site.

https://developer.sony.com/develop/spresense/docs/hw_docs_en.html#_how_to_use_the_external_antenna_for_gnss

Open the GNSS sketch

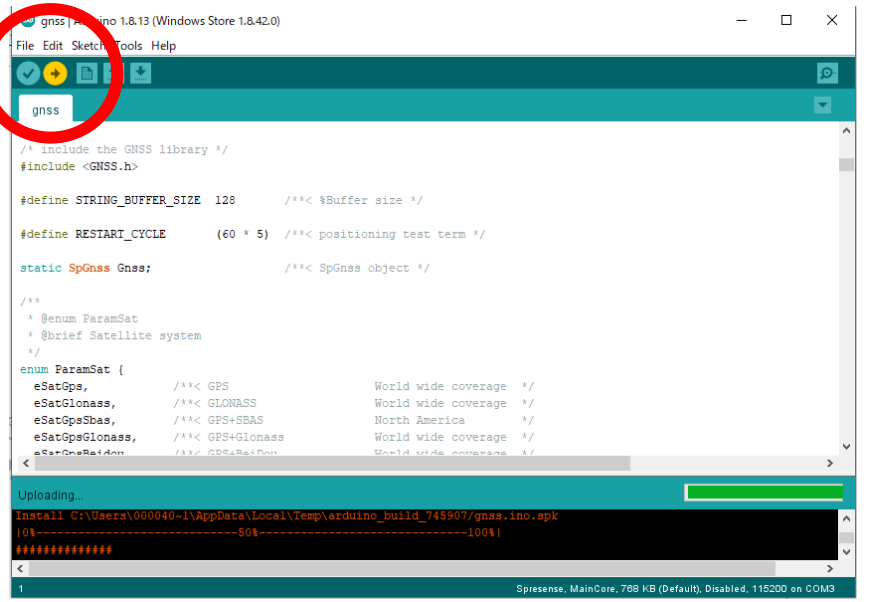
- File → Examples → GNSS → gnss
- You can see the sketch in new window



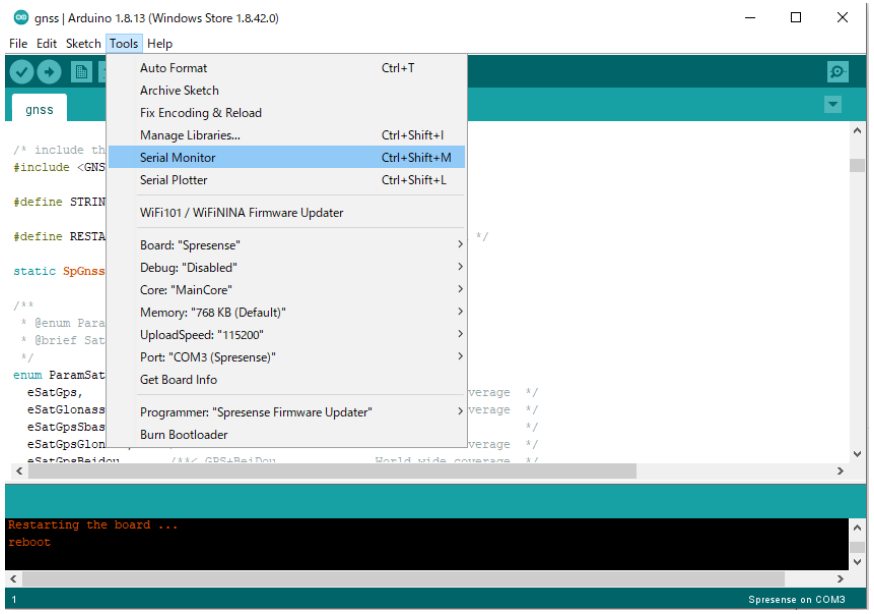


Monitoring

- Upload



- Tool → Serial monitor

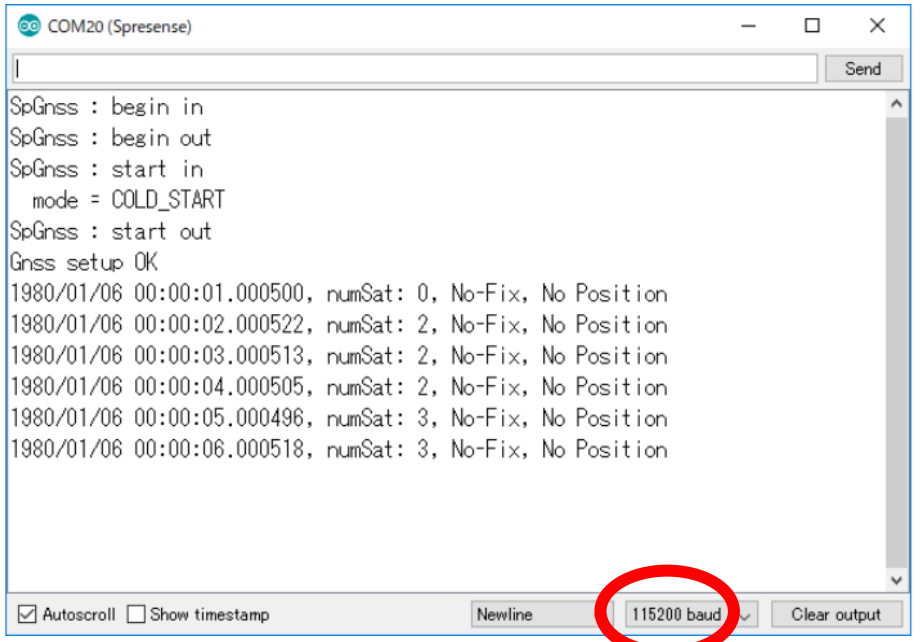


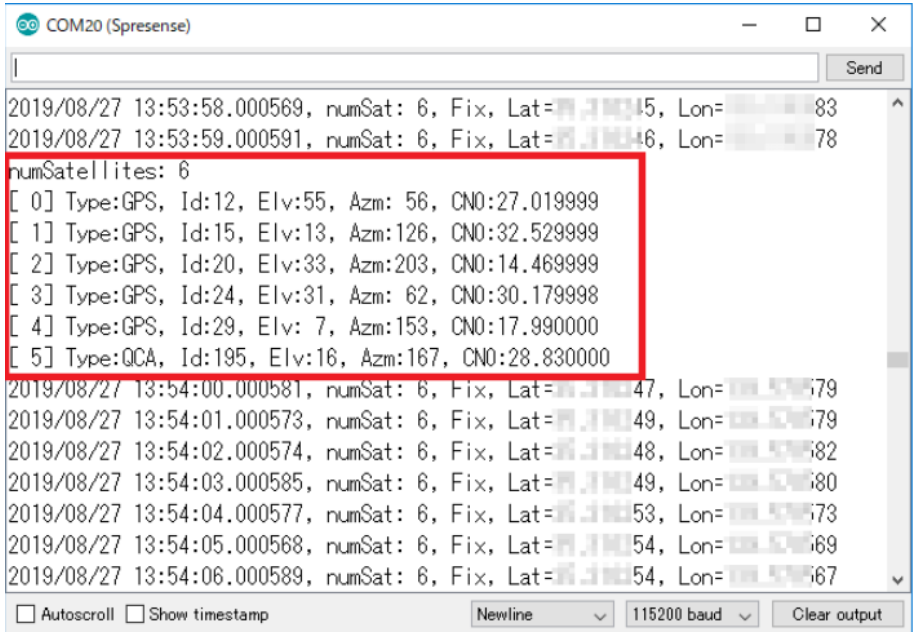
Check the serial monitor

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





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;
```

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

the types of satellite

```
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);
  }
  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);
    }
  }
}
```

Get the result of positioning

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

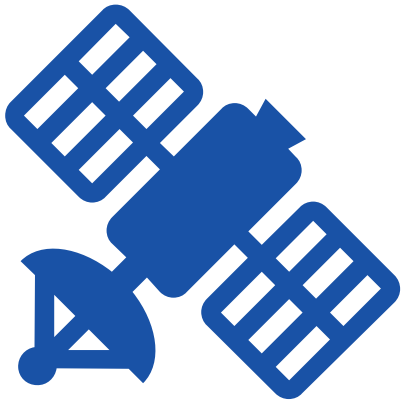
- SpGnss class
  - https://developer.sony.com/develop/spresense/developer-tools/api-reference/api-references-arduino/classSpGnss.html#ade8b8f5d5c05ba2ee67d857a32ef5229
- SpNavData class
  - https://developer.sony.com/develop/spresense/developer-tools/api-reference/api-references-arduino/classSpNavData.html#a29dd2b09d21ed4b7b60a19bce8de01b4

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

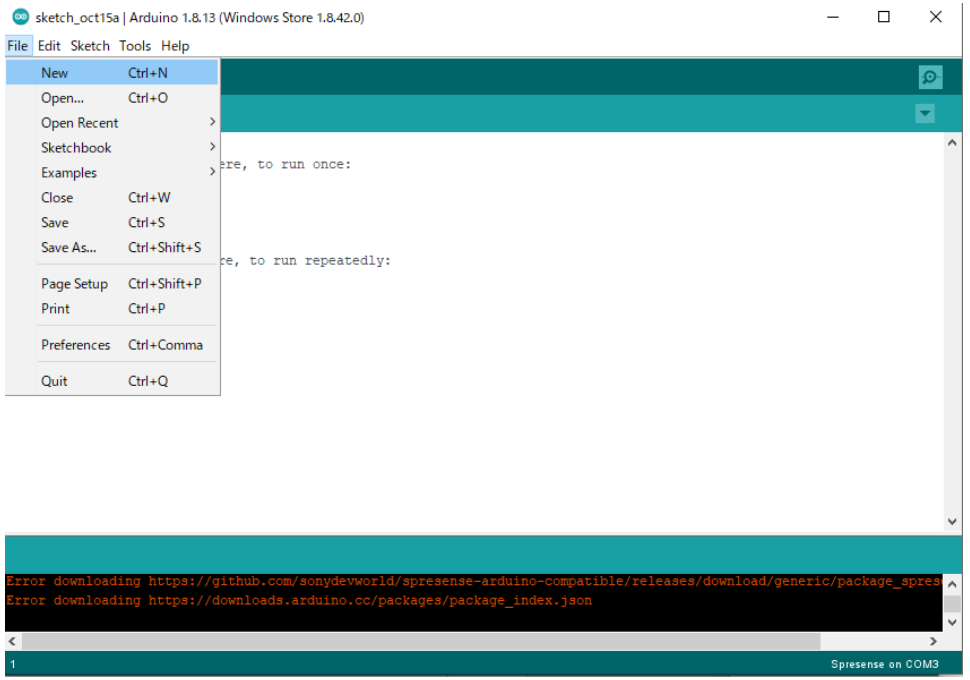
SONY

Tutorial 2

– How to get the disaster information from QZSS in SPRESENSE



- File → New



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

or

- Open QZSS.ino and upload

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

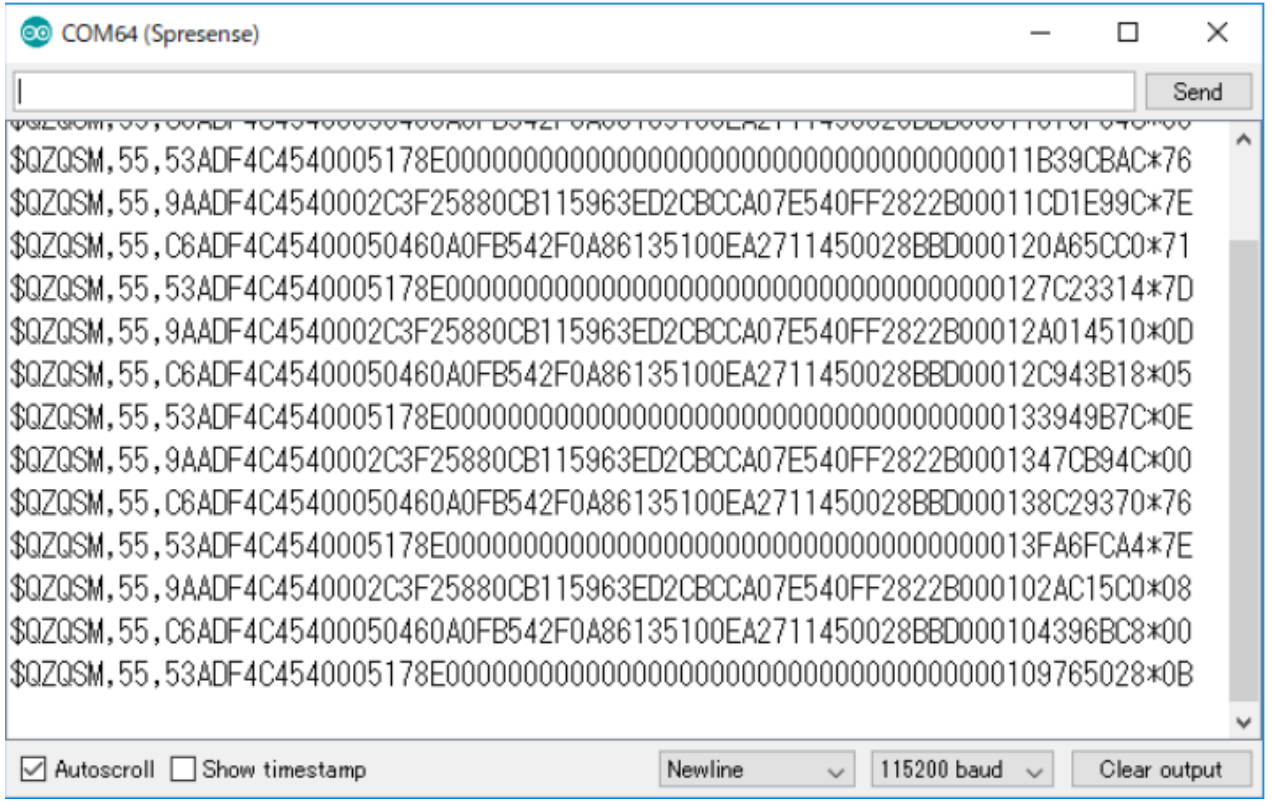
```
  /* use NMEA library */
  NMEA_InitMask();
  NMEA_SetMask(0x4000); // only QZQSM
  NMEA_OUTPUT_CB  funcs;
  funcs.bufReq  = reqbuf;
  funcs.out     = outnmea;
  funcs.outBin  = outbin;
  funcs.bufFree = freebuf;
  NMEA_RegistOutputFunc(&funcs);
}

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



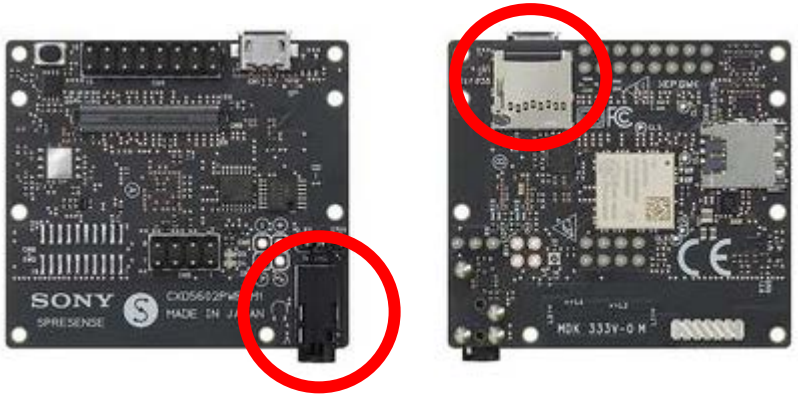
SONY

Tutorial 3

– Play the recorded sound

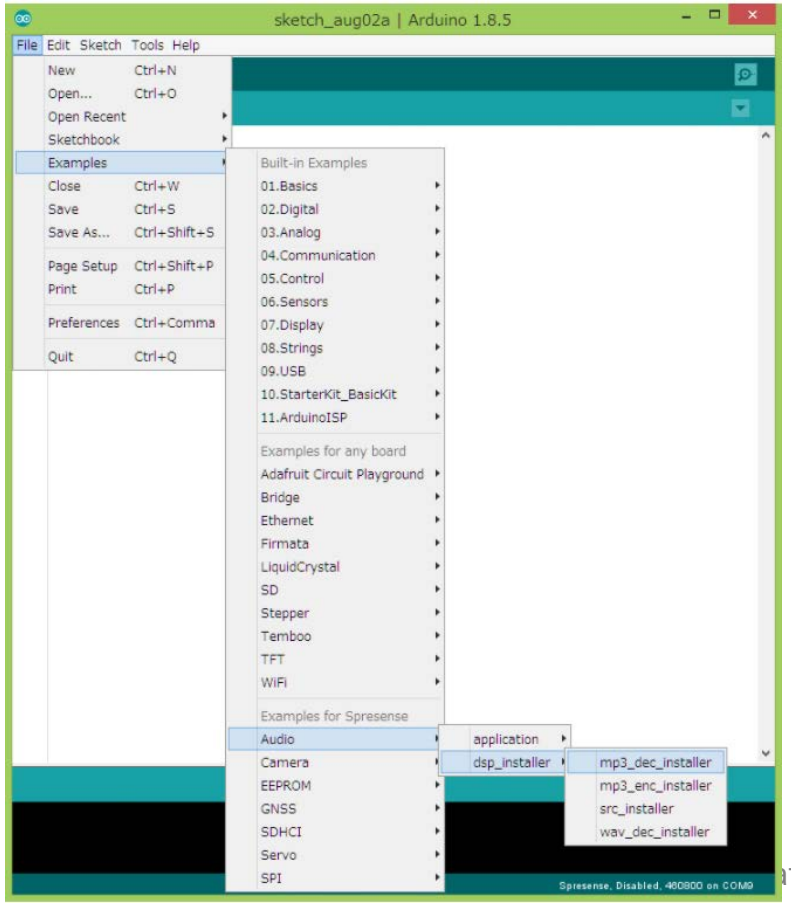
Set up

- 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



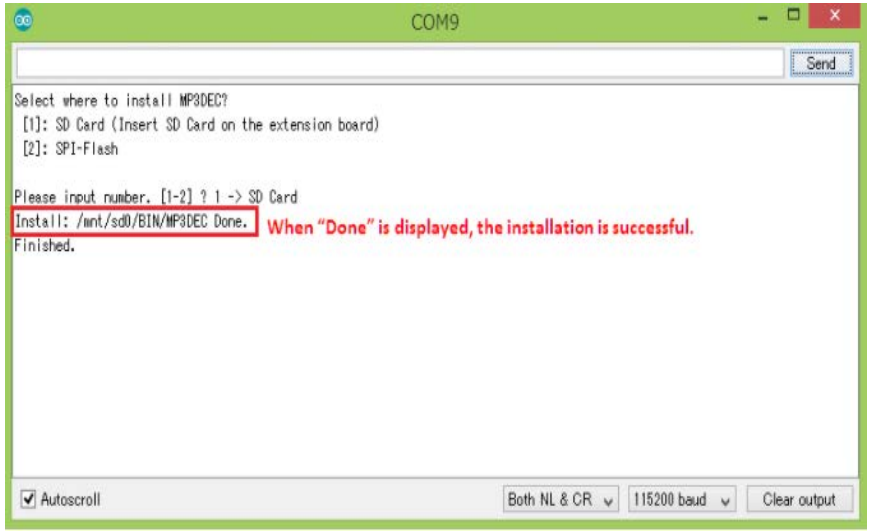
Install MP3 Decoder to SPRESENSE

STEP 1
Select “mp3_dec_installer” and upload



STEP 2
Select install type and baud rate

STEP3
Check if the installation is done



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);

  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);
}
```

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

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);

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

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

Read audio data from SD card and write it to FIFO

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

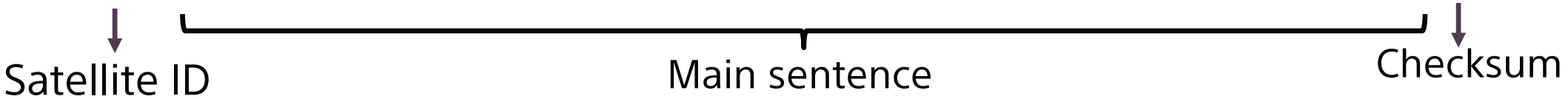
SONY

Demonstration

– Detailed explanation of the demonstration

Decode of QZQSM sentence

- Structure of QZQSM sentence
  - $QZQSM,61,C6ADE568D68006CC030030480438500E2280F154370000000000000138D1393C*0D

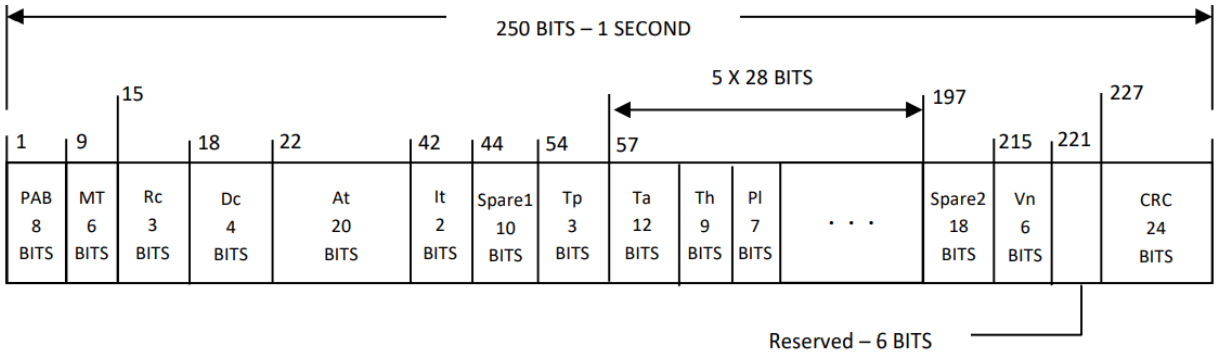


- Change the main sentence from hexadecimal number to binary number
  - 1100011010101101111001010110100011010110100000000000011011001100000000110000010010000000010000111000010100000000111000100010100000001111000101010100001101111000000000000000000000000000000000000000000000100111000110100010011100100111100 (~ 250 bits)

Decode of QZQSM sentence

- Translate with the specification of QZQSM
  - https://qzss.go.jp/technical/download/is_qzss_dcr_008_agree.html
  - 1100011010101101111100101011010001101011010000000000011011001100000000110000000001100001001000000001000011100001010000000011100010001010000000111000101010100001101110000000000000000000000000000000000000000000000000010011100011010001001110010011110

  - E.g. bit 18 ~ 21 means Disaster Category → 1100 (binary) → 12 (decimal) → Typhoon



Disaster Category Code	Description
1	JMA-DC Report (Earthquake Early Warning)
2	JMA-DC Report (Hypocenter)
3	JMA-DC Report (Seismic Intensity)
4	JMA-DC Report (Nankai Trough Earthquake)
5	JMA-DC Report (Tsunami)
6	JMA-DC Report (Northwest Pacific Tsunami)
7	Unused
8	JMA-DC Report (Volcano)
9	JMA-DC Report (Ash Fall)
10	JMA-DC Report (Weather)
11	JMA-DC Report (Flood)
12	JMA-DC Report (Typhoon)
13	Unused
14	JMA-DC Report (Marine)

Sketch of decode

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

SpGnss Gnss;
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;
    }
}


char   PositionData[sizeof(GnssPositionData)];
/* output NMEA */
static char nmea_buf[NMEA_SENTENCE_MAX_LEN];
```

Same with tutorial 3

```
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;
}

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

Same with tutorial 2

```
int get_val(const uint8_t *bytes, int startbit, int bitwidth) {
  int val = 0;
  int index = (startbit + bitwidth - 1) / 8;
  int lsb = 7 - (startbit + bitwidth - 1) % 8;

  int i;
  for (i = 0; i < bitwidth; i++, lsb++) {
    if (lsb > 7) {
      index -= 1;
      lsb = 0;
    }
    val |= ((bytes[index] >> lsb) & 1) << i;
  }
  return val;
}
```

Bring out the part of QZQSM sentence written as binary number and change to decimal number

```
static void play_audio(char *audiofile) {
  /* Open file placed on SD card */
  myFile = theSD.open(audiofile);

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

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

play_audio: Play the recoded sound

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

  theAudio->setVolume(5);
  theAudio->startPlayer(AudioClass::Player0);

  while (1) {
    int err = theAudio->writeFrames(AudioClass::Player0, myFile);

    if (err == AUDIOLIB_ECODE_FILEEND) {
        printf("Main player File End!¥n");
        break;
      }
    /* Show error code from player and stop */
    if (err) {
        printf("Main player error code: %d¥n", err);
        break;
      }
    if (ErrEnd) {
        printf("Error End¥n");
        break;
      }
    usleep(1);
  }

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

Same with tutorial 3

```
void setup() {
  /* Initialize Serial */
  Serial.begin(115200);
  /* Initialize GNSS */
  if (Gnss.begin()) {
    Serial.println("begin error!");
  }

  // start audio system
  theAudio = AudioClass::getInstance();
  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);

err_t err = theAudio->initPlayer(AudioClass::Player0, AS_CODECTYPE_MP3,

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

  /* 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!");
  }
```

Same with tutorial 2, 3

```
    /* 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);

  while (!theSD.begin()) {
    Serial.println("Insert SD card.");
  }
}

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

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

      uint8_t *qzqsm_bytes = ((struct cxd56_gnss_dcreport_data_s*)handle)->sf;
      int message_type = get_val(qzqsm_bytes, 17, 4);
      if (message_type == 12){
        puts("typhoon! Play Alert !");
        play_audio("Sine.mp3");
      }
    }
  }
}
```

qzqsm_bytes is the QZQSM sentences you got as binary number

You can set the processing for each disaster here

Let's do it by yourself outside when you get SPRESENSE

References

- Spresense
  https://developer.sony.com/develop/spresense/docs/introduction_en.html
- Spresense start guide
  https://developer.sony.com/develop/spresense/docs/arduino_set_up_en.html
- Spresense example & tutorial
  https://developer.sony.com/develop/spresense/docs/arduino_tutorials_en.html
- QZSS
  https://qzss.go.jp/en/overview/services/sv02_why.html

SONY

Q & A