Reyax RYS8839 NMEA parsing and Logging

Overview

The Reyax 8839 is a multi bandwidth GNSS reciever. It uses the L1 and L5 band.

Objective

Create a GPS data logger utilizing the multi bandwidth capabilites of the Reyax 8839.

Figure 1 - circuit

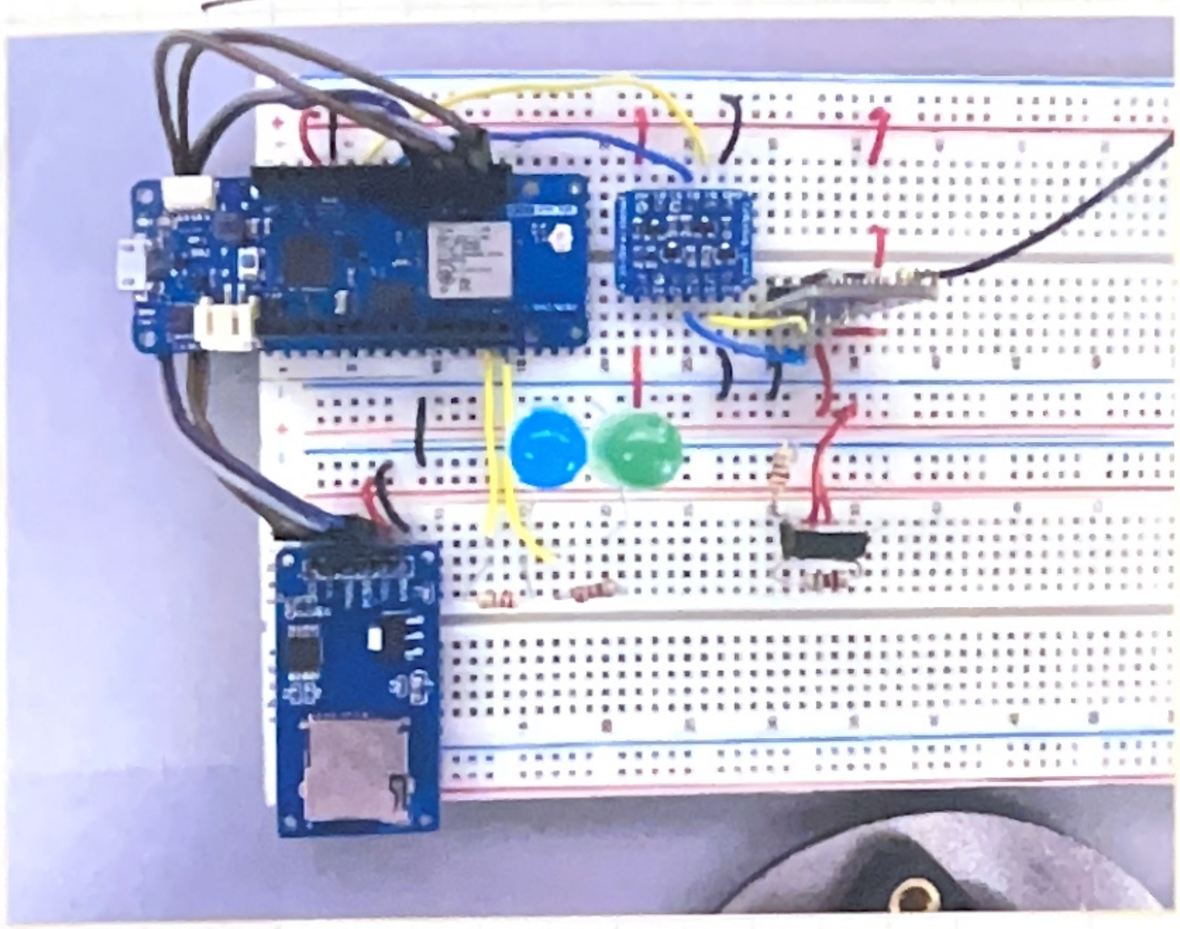


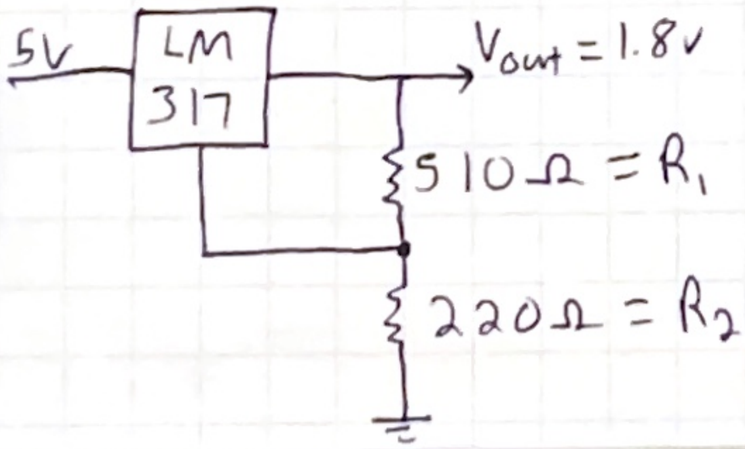
Figure 2 - Antenna

Circuit

The SD card reader is connected via SPI.

The Reyax operates on 1.8 v. To ensure the Rx and Tx of the MKR doesn't damage it, we need a level shifter.

The 1.8V for the level shifter is provided by a LM 317. The power for the 1310 came from an ANKER battery pack.



$$V_{out} = 1.25\left(1 + \frac{R_2}{R_1}\right)$$

$$V_{out} = 1.25\left(1 + \frac{220}{510}\right) = 1.78\,V$$

Reyax Rys 8839 NMEA Parsing and Logging

06/21/2023
Wesley Coota

The Reyax module uses Commands Similar to AT commands. Here are som examples from their Software Guide: reyax.com/products/RYS 8839

3.3.51 @GSTP: Positioning stop

This command is used to stop the positioning. The RYS8839 transfers to the Idle state.

Format: @GSTP<CR><LF>

Argument: None

Response:

Sentence	Description
"[GSTP] Done"	This indicates that the command has been executed successfully.
"[GSTP] Err n"	This indicates that an error has occurred. "n" is where the error code is entered.

3.3.67 @SBSP: SBAS satellite select

This command is used to select which SBAS satellite to be used when SBAS is used. Only one satellite from

GAGAN, WAAS, EGNOS, MSAS, SDCM and BDSBAS can be selected. The RYS8839 will search just the specified SBAS satellite and not search the other SBAS satellites.

Format: @SBSP <arg 1><CR><LF>

Argument:

Field	Description
arg 1	The SBAS satellite is specified. 0: GAGAN (default) 1: WAAS 2: EGNOS 3: MSAS 4: SDCM 5: BDSBAS

Response:

Sentence	Description
"[SBSP] Done"	This indicates that the command has been executed successfully.
"[SBSP] Err n"	This indicates that an error has occurred. "n" is where the error code is entered.

3.3.37 @GNS: Positioning-use satellite setting

This command is used to select the satellite systems to be used for positioning.

The satellite systems are assigned to the bits of the argument. "1" is set for the bits of the systems which are to be used and "0" is set for the bits of the systems which are not be used. Arguments can be specified in decimal or hexadecimal notation. With hexadecimal notation, add "0x" in front of the numeral.

This command must be issued at "Idle" mode.

Format: `@GNS <arg 1><CR><LF>`

Argument:

Field	Description
arg 1	The satellite systems used for positioning are set on a bit by bit basis (0: system not used, 1: system used). bit 0: GPS L1-C/A bit 1: GLONASS L1OF bit 2: SBAS bit 3: QZSS L1-C/A bit 5: QZSS L1-S bit 6: BeiDou B1I bit 7: Galileo E1B/C bit 8: GPS L5 bit 9: QZSS L5 bit 10: BeiDou B1C bit 11: BeiDou B2a bit 12: Galileo E5a bit 13: NavIC (Default value: 0x01)

Response:

Sentence	Description
"[GNS] Done"	This indicates that the command has been executed successfully.
"[GNS] Err n"	This indicates that an error has occurred. "n" is where the error code is entered.

3.3.33 @GGNS: Acquire the positioning-use satellite setting

This command is used to acquire the positioning-use satellite systems setting by @GNS command. It returns the argument of @GNS.

Format: `@GGNS<CR><LF>`

Argument: None

Response:

Sentence	Description
"[GGNS] Done"	This indicates that the command has been executed successfully.
"[GGNS] Err n"	This indicates that an error has occurred. "n" is where the error code is entered.

3.3.5 @BSSL: Output sentence select

This command is used to select the NMEA sentence to be output.

The sentences are assigned to each of the bits of the argument. "1" is set for the bits of the sentences which are to be output, and "0" is set for the bits of the sentences whose output is not required. Arguments can be specified in decimal or hexadecimal notation. With hexadecimal notation, add '0x' in front of the numeral.

Format: `@BSSL <arg 1><CR><LF>`

Argument:

Field	Description
arg 1	Output NMEA sentence bit0 : GGA bit1 : GLL bit2 : GSA bit3 : GSV bit4 : GNS bit5 : RMC bit6 : VTG bit7 : ZDA bit8 : Reserved bit9 : Reserved bit10 : Reserved bit11 : Reserved bit12 : Reserved bit13 : Reserved bit14 : Reserved bit15 : Reserved bit16 : Reserved bit17 : Reserved bit18 : Reserved bit19 : Reserved bit20 : GST (Default value: 0x000000EF)

Response:

Sentence	Description
"[BSSL] Done"	This indicates that the command has been executed successfully.
"[BSSL] Err n"	This indicates that an error has occurred.

3.3.13 @GCD: Cold start

This command is used to start the positioning with cold start. Ephemeris and almanac are erased. Different from @GDCD, time, receiver position and TCXO offset are not erased.

Format: `@GCD<CR><LF>`

Argument: None

Response:

Sentence	Description
"[GCD] Done"	This indicates that the command has been executed successfully.
"[GCD] Err n"	This indicates that an error has occurred. "n" is where the error code is entered.

06/21/2023 Wesley Costa

Reyax RYS8839 NMEA Parsing and Logging

3.3.55 @GTIM: Time setting

This command is used to set the time of the receiver in the RYS8839. The UTC time standard is used for the receiver time which employs the format of year, month, day, hours, minutes and seconds.

The receiver position, current time and TCXO offset value are required in order to initiate a hot start so the time must have been set in the RYS8839 prior to hot start using this command.

Format: `@GTIM <arg 1> <arg 2> <arg 3> <arg 4> <arg 5> <arg 6><CR><LF>`

Field	Description
arg 1	This specifies the UTC time (year) using an integer.
arg 2	This specifies the UTC time (month) using an integer.
arg 3	This specifies the UTC time (day) using an integer.
arg 4	This specifies the UTC time (hour) using an integer.
arg 5	This specifies the UTC time (minutes) using an integer.
arg 6	This specifies the UTC time (seconds) using an integer.

Response:

Sentence	Description
"[GTIM] Done"	This indicates that the command has been executed successfully.
"[GTIM] Err n"	This indicates that an error has occurred. "n" is where the error code is entered.

3.3.39 @GPPS: 1PPS output setting

This command is used to control 1PPS output.

When 1PPS output is enabled, timing pulse output is 1s period from 1PPS output port. When "1" is set to the argument, the timing pulse outputs after clock information being received from GNSS. When "2" is set to the argument, the timing pulse outputs always during positioning operation. When "3" is set to the argument, the timing pulse outputs only during position is fixed.

When 1PPS output is disabled, timing pulse does not output from 1PPS output port.

Format: `@GPPS <arg 1><CR><LF>`

Argument:

Field	Description
arg 1	1PPS output control 0 : Disable 1PPS output (default value) 1 : Enable 1PPS output (output after clock information is received) 2 : Enable 1PPS output (output always while positioning operation) 3 : Enable 1PPS output (output only during position fix)

Response:

Sentence	Description
"[GPPS] Done"	This indicates that the command has been executed successfully.
"[GPPS] Err n"	This indicates that an error has occurred. "n" is where the error code is entered.

Reyax Rys883a NMEA Parsing and Logging

06/21/2023 :3
Wesley Costa

Once the cold start command is issued, the reyax module will begin outputting ~~NEMA~~ NMEA strings.

To parse the ~~NEMA~~ NMEA sentences, we will use the ~~<Arduino NMEA~~ <Arduino Nmea Parser.h>

Tools → Library Manager → Search "Arduino NMEA"
Install "107 - Arduino -NMEA - Parser"

```
#include <SD.h>
#include <SPI.h>
#include <ArduinoNmeaParser.h>

int myIndex = 0;           //What index is our string at?
char mySentence[200];      // Our string has up to 200 chars.
const int chipSelect = 7; // For the SPI SD card.
const int sdLED = 4;
const int fixLED = 5;

void onGgaUpdate(nmea::GgaData const gga)
{
  // This method is called inside of the Arduino Nema Parser
  // when there is new gga data.

  // First we check if there were enough satelities
  // and a good enough signal
  if (gga.fix_quality != nmea::FixQuality::Invalid)
  {
    // Turn on an LED so we can see that the fix is good.
    digitalWrite(fixLED, HIGH);

    // Log some important values from the parser.
    File outputfile = SD.open("data.txt", FILE_WRITE);
    if (outputfile)
    {
      outputfile.print(gga.latitude, 6);
      outputfile.print(", ");
      outputfile.print(gga.longitude, 6);
      outputfile.print(", ");
      outputfile.println(gga.num_satellites);
    }
    outputfile.close();
  }
  else
  {
    // If our fix was invalid, turn off the LED.
    digitalWrite(fixLED, LOW);
  }
}

// This is the parser object
// We pass in the function for onGgaUpdate
// so the parser knows what function to call
// when there is new GGA data. Notice that this
// satement needs to be below the definition of the
// function.
ArduinoNmeaParser parser(NULL, onGgaUpdate);

```

```
50 void setup()
51 {
52   // These pins are outputs
53   pinMode(fixLED, OUTPUT);
54   pinMode(sdLED, OUTPUT);
55   pinMode(chipSelect, OUTPUT);
56
57   Serial.begin(115200);  // Connection to Serial monitor
58   Serial1.begin(115200); // Connection to Reyax Board
59
60   // If the sd card fails to init
61   if (!SD.begin(chipSelect))
62   {
63     Serial.println("Card Failed, or not present... Freezing");
64     digitalWrite(sdLED, HIGH); // Show us there was a problem with the SD.
65     while(1);
66   }
67   else { digitalWrite(sdLED, LOW); }
68
69   // Start the file for the data from the parser
70   File outputfile = SD.open("data.txt", FILE_WRITE);
71   outputfile.println("Latitude, Longitude, NumSats");
72   outputfile.close();
73
74   // Start the file for the full nema sentences
75   File outputnema = SD.open("nema.txt", FILE_WRITE);
76   outputnema.println("=====");
77   outputnema.close();
78
79   // Reyax Commands
80
81   // Stop Positioning
82   Serial1.print("@GSTP\r\n");
83   delay(500);
84   readResponse();
85
86   // Turn on WAAS
87   Serial1.print("@SBSP 1\r\n");
88   delay(500);
89   readResponse();
90
91   //Set to use all possible sats
92   // 0x85 -> GPS L1, GPS L5, and SBAS (WAAS) only.
93   Serial1.print("@GNS 0xFF\r\n");
94   delay(500);
95   readResponse();
96
97   // Get the Satelities being used
98   Serial1.print("@GGNS\r\n");
99   delay(500);
100  readResponse();
101
102  // Turn on only GGA sentences 0x1
103  // Turn on GGA, GSA, and RMC 0x25
104  // Turn on GLL 0x02
105  // Turn on all 0xFF
106  Serial1.print("@BSSL 0x000000FF\r\n");
107  delay(500);
108  readResponse();
109
110  // Inject UTC time for hot start
111  Serial1.print("@GTIM 2023 06 28 08 20 00\r\c");
112  delay(500);
113  readResponse();
114
115  // Enable 1PPS (pulse per second) function
116  Serial1.print("GPPS 1\r\c");
117  delay(500);
118  readResponse();
119
120  // Cold Start
121  Serial1.print("@GCD\r\n");
122  delay(500);
123  readResponse();
124 }
```

8839 parsing and Logging | 65

Reyax RYS83

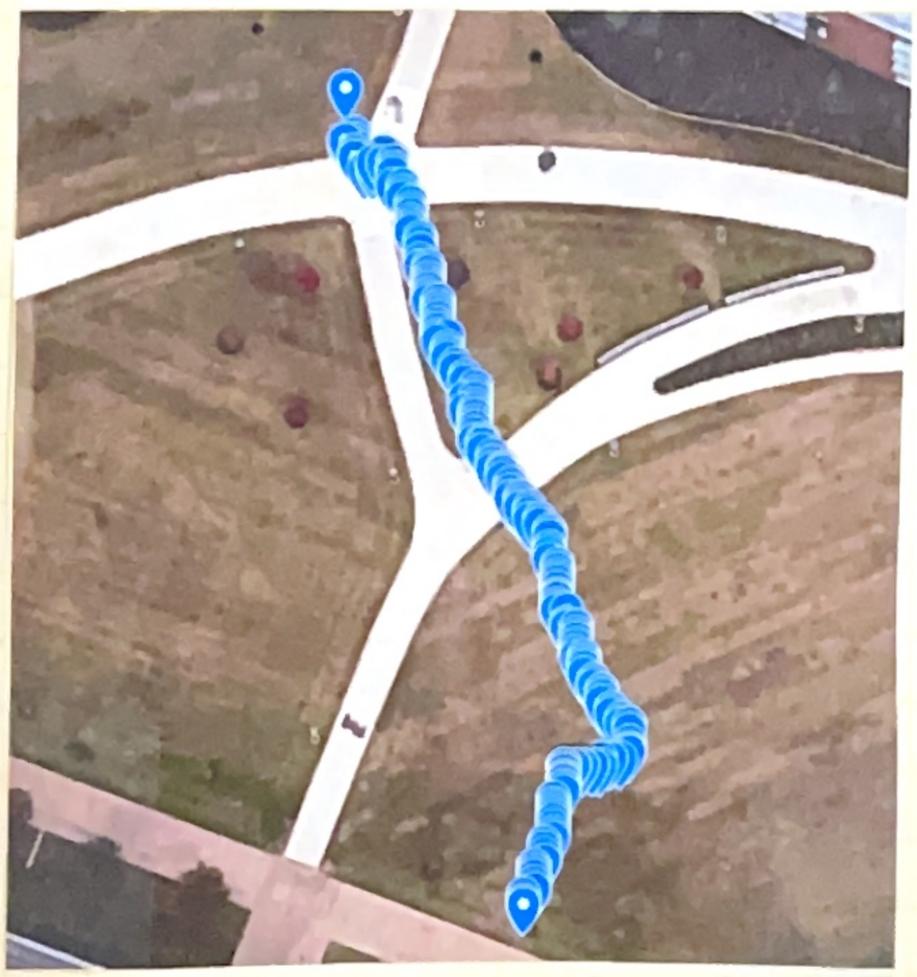
06/21/2023 Written code

Reyax Rys8839 NMEA Parsing and Logging

06/21/2023
Wesley Cooper

```
void readResponse()
{
  // This method will print out the response
  // of the reyax module to the serial monitor

  // A better method would be to save this
  // in a char array and save it to a file
  // so you can debug if something went wrong in the field.

  delay(10);
  while(Serial1.available()>0)
  {
    Serial.print((char)Serial1.read());
  }
}
void loop()
{
  // While we can read something
  while(Serial1.available())
  {
    // Read the char and pass it to the parser
    char car = (char)Serial1.read();
    parser.encode(car);

    // If we see the start of a new sentence
    if (car == '$')
    {
      // Log the last sentence

      // The last char must be a null char.
      mySentence[myIndex] = '\0';

      // Open the nema file we created in setup
      File outputnema = SD.open("nema.txt", FILE_WRITE);
      if (outputnema)
      {
        // Print the sentence
        outputnema.print(mySentence);
      }
      outputnema.close();

      // Get ready for the next sentence
      mySentence[0] = '$';
      mySentence[1] = '\0';
      myIndex = 1;
    }
    else
    {
      // Otherwise, add this char to the end of
      // our current sentence

      mySentence[myIndex] = car;
      myIndex++;
    }
  }
}
```

Reyax RYS8839 NMEA Parsing and Logging



Figures 3 + 4

Example data plotted onto google maps.

Walked around the side walks outside CSM.

Walked into the grass since this area was a problem area with the DFRobot GPS.

```
33.467503, -81.990936, 25
33.467499, -81.990944, 25
33.467495, -81.990944, 25
33.467491, -81.990952, 25
33.467487, -81.990952, 25
33.467484, -81.990959, 25
33.467480, -81.990959, 25
33.467472, -81.990967, 25
33.467468, -81.990967, 25
33.467464, -81.990967, 25
33.467461, -81.990974, 25
33.467457, -81.990974, 25
33.467453, -81.990974, 25
33.467449, -81.990982, 25
```

Figure 5

Example SD card data

Reyax RYS 8839 NMEA Parsing and Logging

06/21/2023
Wesley Cocka

```
$GNGSA,A,3,11,14,27,28,33,41,43,,,,,,1.1,0.6,1.0,4*G,142,40,14,46,039,42,27,26,317,35,28,76,300,50,1*481,47,33,52,330,42,,,,,,,,7*4E
$GNGNS,201637.00,3328.N,0.8,K,A*1C
$GNZDA,201637.00,28,06,2023,,*76
$GPGGA,201638.00,3328.0569,N,08159.4557,W,1,24,0.6,42.0,M,-31.7,M,,*68
$GNGLL,3328.0569,N,08159.4557,W,201638.00,A,A*6D
$GNGSA,A,3,05,11,13,15,18,23,29,,,,,,1.1,0.6,1.0,1*36
$GNGSA,A,3,65,71,72,85,86,87,,,,,,,1.1,0.6,1.0,2*39
$GNGSA,A,3,07,19,26,33,,,,,,,,,1.1,0.6,1.0,3*3F
$GNGSA,A,3,11,14,27,28,33,41,43,,,,,,1.1,0.6,1.0,35,142,39,14,46,039,43,27,26,317,36,28,76,300,50,1*082,47,33,52,330,41,,,,,,,,7*4E
$GNGNS,201638.00,3328N,0.4,K,A*16
$GNZDA,201638.00,28,06,2023,,*79
$GPGGA,201639.00,3328.0567,N,08159.4555,W,1,24,0.6,42.1,M,-31.7,M,,*64
$GNGLL,3328.0567,N,08159.4555,W,201639.00,A,A*60
$GNGSA,A,3,05,11,13,15,18,23,29,,,,,,1.1,0.6,1.0,1*36
$GNGSA,A,3,65,71,72,85,86,87,,,,,,,1.1,0.6,1.0,2*39
$GNGSA,A,3,07,19,26,33,,,,,,,,,1.1,0.6,1.0,3*3F
$GNGSA,A,3,11,14,27,28,33,41,43,,,,,,1.1,0.6,1.0,35,142,38,14,46,039,42,27,26,317,35,28,76,300,50,1*82,47,33,52,330,41,,,,,,,,7*4E
$GNGNS,201639.00,3328.N,0.6,K,A*15
$GNZDA,201639.00,28,06,2023,,*78
$GPGGA,201640.00,3328.0567,N,08159.4555,W,1,24,0.6,42.1,M,-31.7,M,,*6A
$GNGLL,3328.0567,N,08159.4555,W,201640.00,A,A*6E
$GNGSA,A,3,05,11,13,15,18,23,29,,,,,,1.1,0.6,1.0,1*36
$GNGSA,A,3,65,71,72,85,86,87,,,,,,,1.1,0.6,1.0,2*39
$GNGSA,A,3,07,19,26,33,,,,,,,,,1.1,0.6,1.0,3*3F
```

Figure 6

Example NMEA data