CUSRL_Documentation

Generated by Doxygen 1.8.16

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 I2C Namespace Reference

I2C DRIVERS.

Functions

- bool write_reg (uint8_t i2c, uint8_t addr, uint8_t val)
- bool read_regs (uint8_t i2c, uint8_t addr, uint8_t ∗data, uint8_t num)
- bool read_regs (uint8_t i2c, uint8_t ∗data, uint8_t num)

4.1.1 Detailed Description

I2C DRIVERS.

In this namespace the I2C drivers are declared and then defined for use with the I2C protocol

4.1.2 Function Documentation

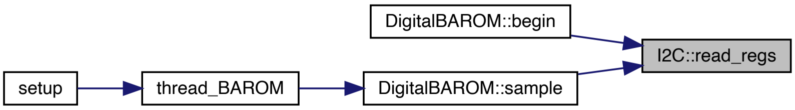
4.1.2.1 read_regs() [1/2]

```
bool I2C::read_regs (
             uint8_t i2c,
             uint8_t * data,
             uint8_t num )
```

4.1.2.2 read_regs() [2/2]

```
bool I2C::read_regs (
            uint8_t i2c,
            uint8_t addr,
            uint8_t * data,
            uint8_t num )
```

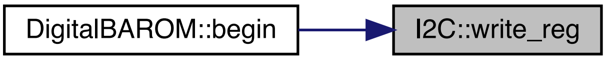
Here is the caller graph for this function:



4.1.2.3 write_reg()

```
bool I2C::write_reg (
            uint8_t i2c,
            uint8_t addr,
            uint8_t val )
```

Here is the caller graph for this function:



4.2 INITS Namespace Reference

Variables

- int speakerPin = 2
- int highG_xPin = 33
- int highG_yPin = 34
- int highG_zPin = 35
- DigitalIMU IMU = DigitalIMU(55,0x28)
- DigitalBAROM BAROM
- AnalogIMU HIGHG = AnalogIMU(highG_xPin,highG_yPin,highG_zPin,true)
- BeepyBOI berp = BeepyBOI(speakerPin)
- DLLflash * flash
- GPSdata gps_data
- IMUdata imu_data
- BAROMdata barom_data
- ACCELdata accel_data

4.2.1 Detailed Description

All pointers and objects that are required in the main file are put within the INITS namespace to avoid global variables and their implications

4.2.2 Variable Documentation

4.2.2.1 accel_data

```
ACCELdata INITS::accel_data
```

4.2.2.2 BAROM

```
DigitalBAROM INITS::BAROM
```

4.2.2.3 barom_data

```
BAROMdata INITS::barom_data
```

4.2.2.4 berp

```
BeepyBOI INITS::berp = BeepyBOI(speakerPin)
```

4.2.2.5 flash

```
DLLflash * INITS::flash
```

4.2.2.6 gps_data

```
GPSdata INITS::gps_data
```

4.2.2.7 HIGHG

```
AnalogIMU INITS::HIGHG = AnalogIMU(highG_xPin,highG_yPin,highG_zPin,true)
```

4.2.2.8 highG_xPin

```
int INITS::highG_xPin = 33
```

4.2.2.9 highG_yPin

```
int INITS::highG_yPin = 34
```

4.2.2.10 highG_zPin

```
int INITS::highG_zPin = 35
```

4.2.2.11 IMU

```
DigitalIMU INITS::IMU = DigitalIMU(55,0x28)
```

4.2.2.12 imu_data

```
IMUdata INITS::imu_data
```

4.2.2.13 speakerPin

```
int INITS::speakerPin = 2
```

4.3 PROTOTHREADING Namespace Reference

PROTOTHREADING DEFINITIONS.

Variables

- int interval_IMU = 45

  PROTOTHREADING TIME INTERVALS.
- int interval_BAROM = 2000
- int interval_ACCEL = 50
- ThreadController thread_control = ThreadController()

  PROTOTHREADING Declaration and Definitions.
- Thread ∗ ThreadIMU = new Thread()
- Thread ∗ ThreadBAROM = new Thread()
- Thread ∗ ThreadACCEL = new Thread()

4.3.1 Detailed Description

PROTOTHREADING DEFINITIONS.

THE VARIABLES ARE DEFINED HERE

The protothreading system is implemented by the ArduinoThread library

The Teensy 3.6 / 4.0 micro-controllers are one core, one thread therefore true asynchronous operation cannot be accomplished This is unfortunate because optimally all sampling of the data should be done at the same time Therefore in order to work around this limitation, a form of threading was introduced that approaches asynchronous operation without actually achieving it, aka protothreading

The way it works is essentially by having a(n) overall controller (i.e. ThreadController class) that manages the timing of all functions you want to run Whenever a function has reached the time it needs to be called again the ThreadController will call the the function and interrupt whatever is currently running

The pro about this is that it also allows us to deal with different intervals that sensors or components require Such as one component needing more time over the other Every process interval is based off of the datasheet and its recommendations on sampling time

4.3.2 Variable Documentation

4.3.2.1 interval_ACCEL

```
int PROTOTHREADING::interval_ACCEL = 50
```

4.3.2.2 interval_BAROM

```
int PROTOTHREADING::interval_BAROM = 2000
```

4.3.2.3 interval_IMU

```
int PROTOTHREADING::interval_IMU = 45
```

PROTOTHREADING TIME INTERVALS.

Defining the time intervals (in milliseconds) at which to call the "threads"

Every important task (i.e. sampling, writing to flash, RF, etc...) has a "thread" and their intervals are defined here.

4.3.2.4 thread_control

```
ThreadController PROTOTHREADING::thread_control = ThreadController()
```

PROTOTHREADING Declaration and Definitions.

All the required thread objects and pointers are declared and defined here thread_control is the overarching ThreadController that handles all the timing Every thread is a pointer that is pointing to an object, or instance, of the Thread class initiated dynimically in order to use them in any scope

4.3.2.5 ThreadACCEL

```
Thread * PROTOTHREADING::ThreadACCEL = new Thread()
```

4.3.2.6 ThreadBAROM

```
Thread * PROTOTHREADING::ThreadBAROM = new Thread()
```

4.3.2.7 ThreadIMU

```
Thread * PROTOTHREADING::ThreadIMU = new Thread()
```

Chapter 5

Class Documentation

5.1 ACCELdata Struct Reference

```
#include <StructDefs.hpp>
```

Public Attributes

- float x
- float y
- float z
- uint32_t t

5.1.1 Member Data Documentation

5.1.1.1 t

```
uint32_t ACCELdata::t
```

5.1.1.2 x

```
float ACCELdata::x
```

5.1.1.3 y

```
float ACCELdata::y
```

5.1.1.4 z

```
float ACCELdata::z
```

The documentation for this struct was generated from the following files:

- src/src/StructDefs.hpp
- src/src/yonics.hpp

5.2 AnalogIMU Class Reference

```
#include <yonics.hpp>
```

Public Member Functions

- AnalogIMU ()
- AnalogIMU (int xPin, int yPin, int zPin)
- AnalogIMU (int xPin, int yPin, int zPin, bool highBitDepth)
- void sample (ACCELdata *data)

5.2.1 Constructor & Destructor Documentation

5.2.1.1 AnalogIMU() [1/3]

```
AnalogIMU::AnalogIMU ( )
```

5.2.1.2 AnalogIMU() [2/3]

```
AnalogIMU::AnalogIMU (
              int xPin,
              int yPin,
              int zPin )
```

5.2.1.3 AnalogIMU() [3/3]

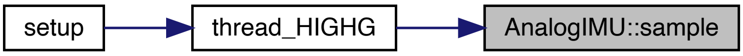
```
AnalogIMU::AnalogIMU (
              int xPin,
              int yPin,
              int zPin,
              bool highBitDepth )
```

5.2.2 Member Function Documentation

5.2.2.1 sample()

```
void AnalogIMU::sample (
            ACCELdata * data )
```

Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- src/src/yonics.hpp
- src/src/AnalogIMU.cpp

5.3 BAROMdata Struct Reference

```
#include <StructDefs.hpp>
```

Public Attributes

- float pressure = 0
- float altitude = 0
- float temperature = 0
- uint32_t t = 0

5.3.1 Member Data Documentation

5.3.1.1 altitude

```
float BAROMdata::altitude = 0
```

5.3.1.2 pressure

```
float BAROMdata::pressure = 0
```

5.3.1.3 t

```
uint32_t BAROMdata::t = 0
```

5.3.1.4 temperature

```
float BAROMdata::temperature = 0
```

The documentation for this struct was generated from the following files:

- src/src/StructDefs.hpp
- src/src/yonics.hpp

5.4 BeepyBOI Class Reference

```
#include <yonics.hpp>
```

Public Member Functions

- BeepyBOI ()
- BeepyBOI (int pin)
- void hello ()
- void error ()
- void countdown (int s)
- void lowBeep ()
- void midBeep ()
- void hiBeep ()
- void bombBeep ()

5.4.1 Constructor & Destructor Documentation

5.4.1.1 BeepyBOI() [1/2]

```
BeepyBOI::BeepyBOI ( )
```

5.4.1.2 BeepyBOI() [2/2]

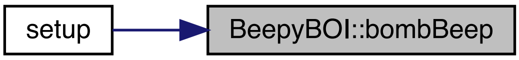
```
BeepyBOI::BeepyBOI (
             int pin )
```

5.4.2 Member Function Documentation

5.4.2.1 bombBeep()

```
void BeepyBOI::bombBeep ( )
```

Here is the caller graph for this function:



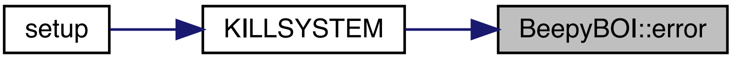
5.4.2.2 countdown()

```
void BeepyBOI::countdown (
             int s )
```

5.4.2.3 error()

```
void BeepyBOI::error ( )
```

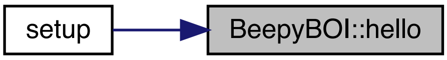
Here is the caller graph for this function:



5.4.2.4 hello()

```
void BeepyBOI::hello ( )
```

Here is the caller graph for this function:



5.4.2.5 hiBeep()

```
void BeepyBOI::hiBeep ( )
```

5.4.2.6 lowBeep()

```
void BeepyBOI::lowBeep ( )
```

5.4.2.7 midBeep()

```
void BeepyBOI::midBeep ( )
```

The documentation for this class was generated from the following files:

- src/src/yonics.hpp
- src/src/BeepyBOI.cpp

5.5 DigitalBAROM Class Reference

```
#include <yonics.hpp>
```

Public Member Functions

- DigitalBAROM ()
- bool begin ()
- bool sample (BAROMdata ∗data)

5.5.1 Constructor & Destructor Documentation

5.5.1.1 DigitalBAROM()

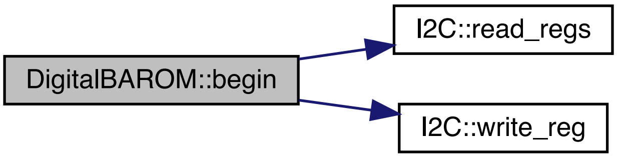
```
DigitalBAROM::DigitalBAROM ( )
```

5.5.2 Member Function Documentation

5.5.2.1 begin()

```
bool DigitalBAROM::begin ( )
```

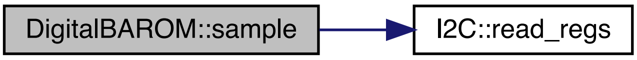
Here is the call graph for this function:



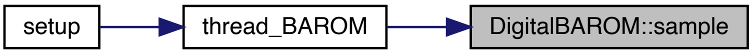
5.5.2.2 sample()

```
bool DigitalBAROM::sample (
            BAROMdata * data )
```

Here is the call graph for this function:



Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- src/src/yonics.hpp
- src/src/DigitalBAROM.cpp

5.6 DigitalGPS Class Reference

```
#include <yonics.hpp>
```

Public Member Functions

- DigitalGPS (HardwareSerial ∗ser)
- void dummyPrint ()
- void initGPS ()
- void eraseLOCUS ()
- void GPSData_dump_setup ()
- void refresh_GPSData (bool gpsecho)
- void pullGPSFlashData ()
- void pullRawGPS ()

Public Attributes

- Adafruit_GPS ∗ GPS
- HardwareSerial ∗ GPSSerial

5.6.1 Constructor & Destructor Documentation

5.6.1.1 DigitalGPS()

```
DigitalGPS::DigitalGPS (
            HardwareSerial ∗ ser )
```

5.6.2 Member Function Documentation

5.6.2.1 dummyPrint()

```
void DigitalGPS::dummyPrint ( )
```

5.6.2.2 eraseLOCUS()

```
void DigitalGPS::eraseLOCUS ( )
```

5.6.2.3 GPSData_dump_setup()

```
void DigitalGPS::GPSData_dump_setup ( )
```

5.6.2.4 initGPS()

```
void DigitalGPS::initGPS ( )
```

5.6.2.5 pullGPSFlashData()

```
void DigitalGPS::pullGPSFlashData ( )
```

5.6.2.6 pullRawGPS()

```
void DigitalGPS::pullRawGPS ( )
```

5.6.2.7 refresh_GPSData()

```
void DigitalGPS::refresh_GPSData (
            bool gpsecho )
```

5.6.3 Member Data Documentation

5.6.3.1 GPS

```
Adafruit_GPS* DigitalGPS::GPS
```

5.6.3.2 GPSSerial

```
HardwareSerial* DigitalGPS::GPSSerial
```

The documentation for this class was generated from the following files:

- src/src/yonics.hpp
- src/src/DigitalGPS.cpp

5.7 DigitalIMU Class Reference

```
#include <yonics.hpp>
```

Public Member Functions

- DigitalIMU ()
- DigitalIMU (int32_t sensorID, uint8_t address)
- bool begin ()
- void sample (IMUdata *data)

5.7.1 Constructor & Destructor Documentation

5.7.1.1 DigitalIMU() [1/2]

```
DigitalIMU::DigitalIMU ( )
```

5.7.1.2 DigitalIMU() [2/2]

```
DigitalIMU::DigitalIMU (
            int32_t sensorID,
            uint8_t address )
```

5.7.2 Member Function Documentation

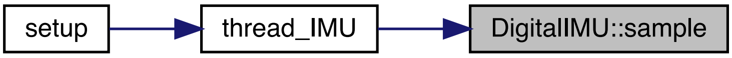
5.7.2.1 begin()

```
bool DigitalIMU::begin ( )
```

5.7.2.2 sample()

```
void DigitalIMU::sample (
             IMUdata * data )
```

Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- src/src/yonics.hpp
- src/src/DigitalIMU.cpp

5.8 DLLflash Class Reference

```
#include <DLLflash.hpp>
```

Public Member Functions

- DLLflash ()
- DLLflash (int)
- ∼DLLflash ()
- template<class T >
  void addType (T ∗, char ∗)
- bool writeSample (char ∗)
- void setToWrite ()
- void setToRead ()

5.8.1 Constructor & Destructor Documentation

5.8.1.1 DLLflash() [1/2]

```
DLLflash::DLLflash ( )
```

5.8.1.2 DLLflash() [2/2]

```
DLLflash::DLLflash (
             int flashpin )
```

5.8.1.3 ∼DLLflash()

```
DLLflash::∼DLLflash ( )
```

5.8.2 Member Function Documentation

5.8.2.1 addType()

```
template<class T >
void DLLflash::addType (
             T * data,
             char * id )
```

5.8.2.2 setToRead()

```
void DLLflash::setToRead ( )
```

5.8.2.3 setToWrite()

```
void DLLflash::setToWrite ( )
```

5.8.2.4 writeSample()

```
bool DLLflash::writeSample (
             char *  )
```

The documentation for this class was generated from the following files:

- src/src/DLLflash.hpp
- src/src/DLLflash.cpp

5.9 DLLtype Class Reference

```
#include <DLLflash.hpp>
```

Public Member Functions

- DLLtype (void ∗, int, char ∗)
- ∼DLLtype ()
- bool setType (void ∗, int)
- bool writeSample (uint32_t, SPIFlash ∗)
- bool readSample ()
- char ∗ getID ()

5.9.1 Constructor & Destructor Documentation

5.9.1.1 DLLtype()

```
DLLtype::DLLtype (
             void * dataPtr,
             int dataSize,
             char * id )
```

5.9.1.2 ∼DLLtype()

```
DLLtype::~DLLtype ( )
```

5.9.2 Member Function Documentation

5.9.2.1 getID()

```
char * DLLtype::getID ( )
```

5.9.2.2 readSample()

```
bool DLLtype::readSample ( )
```

5.9.2.3 setType()

```
bool DLLtype::setType (
             void * dataPtr,
             int dataSize )
```

5.9.2.4 writeSample()

```
bool DLLtype::writeSample (
             uint32_t next,
             SPIFlash * flash )
```

The documentation for this class was generated from the following files:

- src/src/DLLflash.hpp
- src/src/DLLflash.cpp

5.10 GPSdata Struct Reference

```
#include <StructDefs.hpp>
```

Public Attributes

- float lat = 0
- float lon = 0
- float altitude = 0
- float speed = 0
- float angle = 0
- float sat_num = 0
- uint32_t t

5.10.1 Member Data Documentation

5.10.1.1 altitude

```
float GPSdata::altitude = 0
```

5.10.1.2 angle

```
float GPSdata::angle = 0
```

5.10.1.3 lat

```
float GPSdata::lat = 0
```

5.10.1.4 lon

```
float GPSdata::lon = 0
```

5.10.1.5 sat_num

```
float GPSdata::sat_num = 0
```

5.10.1.6 speed

```
float GPSdata::speed = 0
```

5.10.1.7 t

```
uint32_t GPSdata::t
```

The documentation for this struct was generated from the following files:

- src/src/StructDefs.hpp
- src/src/yonics.hpp

5.11 IMUdata Struct Reference

```
#include <StructDefs.hpp>
```

Public Attributes

- double orient_euler [3] = {0,0,0}
- double gyro_fused [3] = {0,0,0}
- double accel_fused [3] = {0,0,0}
- double accel_raw [3] = {0,0,0}
- double gyro_raw [3] = {0,0,0}
- double magnetometer [3] = {0,0,0}
- double orient_quat [4] = {0,0,0,0}
- uint32_t t = 0

5.11.1 Member Data Documentation

5.11.1.1 accel_fused

```
double IMUdata::accel_fused = {0,0,0}
```

5.11.1.2 accel_raw

```
double IMUdata::accel_raw = {0,0,0}
```

5.11.1.3 gyro_fused

```
double IMUdata::gyro_fused = {0,0,0}
```

5.11.1.4 gyro_raw

```
double IMUdata::gyro_raw = {0,0,0}
```

5.11.1.5 magnetometer

```
double IMUdata::magnetometer = {0,0,0}
```

5.11.1.6 orient_euler

```
double IMUdata::orient_euler = {0,0,0}
```

5.11.1.7 orient_quat

```
double IMUdata::orient_quat = {0,0,0,0}
```

5.11.1.8 t

```
uint32_t IMUdata::t = 0
```

The documentation for this struct was generated from the following files:

- src/src/StructDefs.hpp
- src/src/yonics.hpp

5.12 RFM96W_Client Class Reference

```
#include <RFM96W.hpp>
```

Public Member Functions

- RFM96W_Client (uint8_t slaveSelectPin, uint8_t interruptPin, RHHardwareSPI hardware_spi)
- ∼RFM96W_Client ()
- void init ()
- void loop (uint8_t ∗data)
- void TX (const uint8_t ∗data)
- void RX ()

Public Attributes

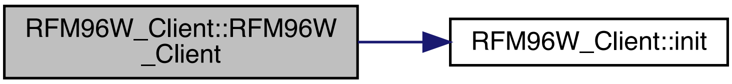
- RH_RF95 ∗ Client_Driver
- RHReliableDatagram ∗ Client_Manager

5.12.1 Constructor & Destructor Documentation

5.12.1.1 RFM96W_Client()

```
RFM96W_Client::RFM96W_Client (
             uint8_t slaveSelectPin,
             uint8_t interruptPin,
             RHHardwareSPI hardware_spi )
```

Here is the call graph for this function:



5.12.1.2 ∼RFM96W_Client()

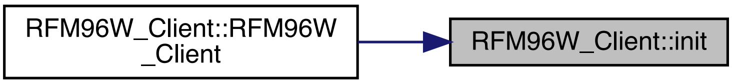
```
RFM96W_Client::∼RFM96W_Client ( )
```

5.12.2 Member Function Documentation

5.12.2.1 init()

```
void RFM96W_Client::init ( )
```

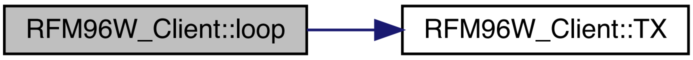
Here is the caller graph for this function:



5.12.2.2 loop()

```
void RFM96W_Client::loop (
            uint8_t * data )
```

Here is the call graph for this function:



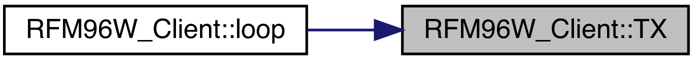
5.12.2.3 RX()

```
void RFM96W_Client::RX ( )
```

5.12.2.4 TX()

```
void RFM96W_Client::TX (
            const uint8_t * data )
```

Here is the caller graph for this function:



5.12.3 Member Data Documentation

5.12.3.1 Client_Driver

```
RH_RF95* RFM96W_Client::Client_Driver
```

5.12.3.2 Client_Manager

```
RHReliableDatagram* RFM96W_Client::Client_Manager
```

The documentation for this class was generated from the following files:

- src/src/RFM96W.hpp
- src/src/RFM96W.cpp

5.13 RFM96W_Server Class Reference

```
#include <RFM96W.hpp>
```

Public Member Functions

- RFM96W_Server (uint8_t slaveSelectPin, uint8_t interruptPin, RHHardwareSPI hardware_spi)
- ∼RFM96W_Server ()
- void init ()
- void loop ()
- void TX ()
- void RX ()

Public Attributes

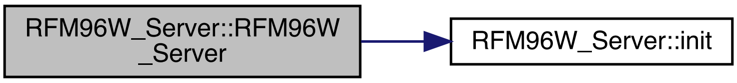
- RH_RF95 ∗ Server_Driver
- RHReliableDatagram ∗ Server_Manager

5.13.1 Constructor & Destructor Documentation

5.13.1.1 RFM96W_Server()

```
RFM96W_Server::RFM96W_Server (
            uint8_t slaveSelectPin,
            uint8_t interruptPin,
            RHHardwareSPI hardware_spi )
```

Here is the call graph for this function:



5.13.1.2 ∼RFM96W_Server()

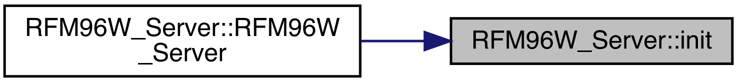
```
RFM96W_Server::∼RFM96W_Server ( )
```

5.13.2 Member Function Documentation

5.13.2.1 init()

```
void RFM96W_Server::init ( )
```

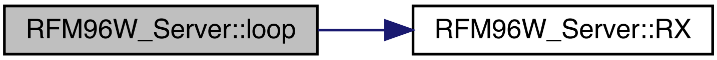
Here is the caller graph for this function:



5.13.2.2 loop()

```
void RFM96W_Server::loop ( )
```

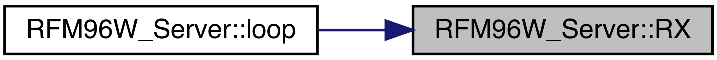
Here is the call graph for this function:



5.13.2.3 RX()

```
void RFM96W_Server::RX ( )
```

Here is the caller graph for this function:



5.13.2.4 TX()

```
void RFM96W_Server::TX ( )
```

5.13.3 Member Data Documentation

5.13.3.1 Server_Driver

```
RH_RF95* RFM96W_Server::Server_Driver
```

5.13.3.2 Server_Manager

```
RHReliableDatagram* RFM96W_Server::Server_Manager
```

The documentation for this class was generated from the following files:

- src/src/RFM96W.hpp
- src/src/RFM96W.cpp

5.14 SaveSD Class Reference

```
#include <yonics.hpp>
```

Public Member Functions

- SaveSD ()
- bool initFolder ()
- bool sampleIMU (IMUdata ∗data)
- bool sampleBAROM (BAROMdata ∗data)
- bool sampleACCEL (ACCELdata ∗data)
- bool sampleGPS (GPSdata ∗data)

5.14.1 Constructor & Destructor Documentation

5.14.1.1 SaveSD()

```
SaveSD::SaveSD ( )
```

5.14.2 Member Function Documentation

5.14.2.1 initFolder()

```
bool SaveSD::initFolder ( )
```

5.14.2.2 sampleACCEL()

```
bool SaveSD::sampleACCEL (
            ACCELdata ∗ data )
```

5.14.2.3 sampleBAROM()

```
bool SaveSD::sampleBAROM (
            BAROMdata ∗ data )
```

5.14.2.4 sampleGPS()

```
bool SaveSD::sampleGPS (
            GPSdata ∗ data )
```

5.14.2.5 sampleIMU()

```
bool SaveSD::sampleIMU (
            IMUdata ∗ data )
```

The documentation for this class was generated from the following files:

- src/src/yonics.hpp
- src/src/SaveSD.cpp

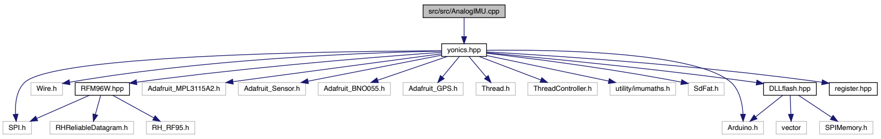
Chapter 6

File Documentation

6.1 src/src/AnalogIMU.cpp File Reference

```
#include "yonics.hpp"
```

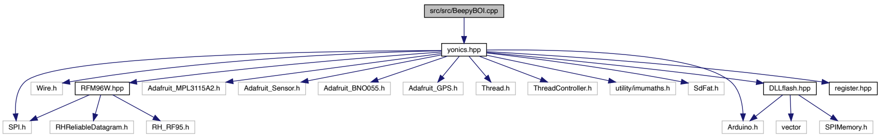
Include dependency graph for AnalogIMU.cpp:



6.2 src/src/BeepyBOI.cpp File Reference

```
#include "yonics.hpp"
```

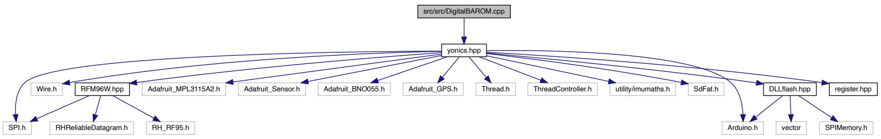
Include dependency graph for BeepyBOI.cpp:



6.3 src/src/DigitalBAROM.cpp File Reference

```
#include "yonics.hpp"
```

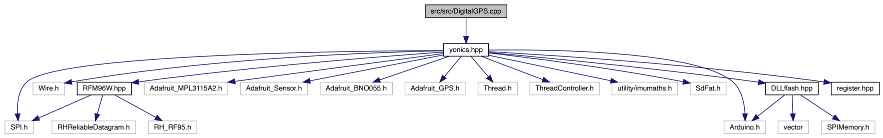
Include dependency graph for DigitalBAROM.cpp:



6.4 src/src/DigitalGPS.cpp File Reference

```
#include <yonics.hpp>
```

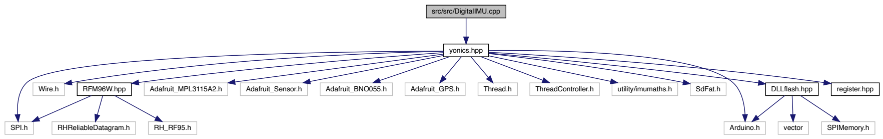
Include dependency graph for DigitalGPS.cpp:



6.5 src/src/DigitalIMU.cpp File Reference

```
#include "yonics.hpp"
```

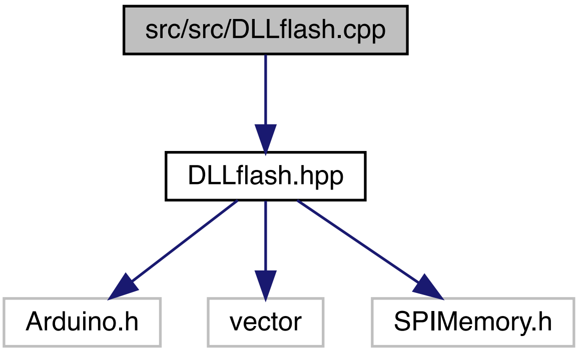
Include dependency graph for DigitalIMU.cpp:



6.6 src/src/DLLflash.cpp File Reference

```
#include "DLLflash.hpp"
```

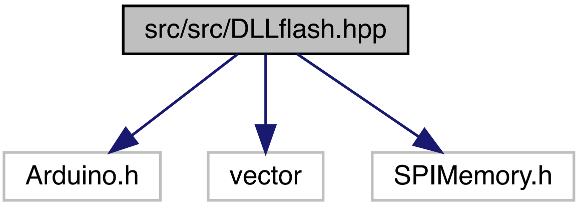
Include dependency graph for DLLflash.cpp:



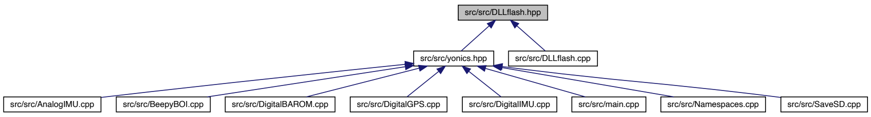
6.7 src/src/DLLflash.hpp File Reference

```
#include <Arduino.h>
#include <vector>
#include <SPIMemory.h>
```

Include dependency graph for DLLflash.hpp:



This graph shows which files directly or indirectly include this file:



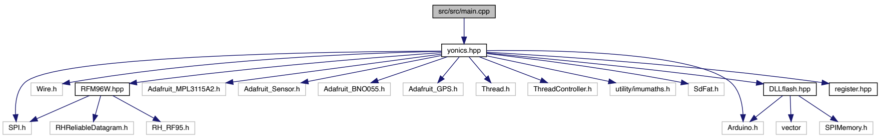
Classes

- class DLLtype
- class DLLflash

6.8 src/src/main.cpp File Reference

```
#include "yonics.hpp"
```

Include dependency graph for main.cpp:



Functions

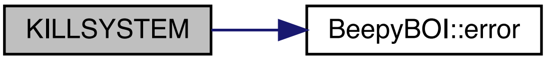
- void thread_IMU ()
- void thread_BAROM ()
- void thread_HIGHG ()
- void KILLSYSTEM ()
- void setup ()
- void loop ()

6.8.1 Function Documentation

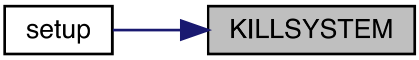
6.8.1.1 KILLSYSTEM()

```
void KILLSYSTEM ( )
```

TO RESET THE SYSTEM POWER MUST BE REMOVED AND THEN REAPPLIEDHere is the call graph for this function:



Here is the caller graph for this function:



6.8.1.2 loop()

```
void loop ( )
```

< Loop the ThreadController

6.8.1.3 setup()

```
void setup ( )
```

< Wait 2.5 seconds before starting everything up

< Start serial comms

< Hello beep

< STILL VERY WIP FLASH INITIALIZATION

Initialize BNO055 IMU sensor

Initialize MPL3115A2 sensor

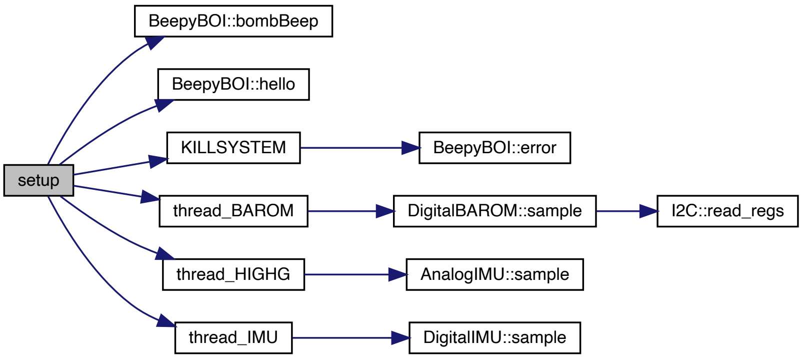
Configure IMU thread

Configure Barometer thread

Configure Accelerometer thread

Add threads to ThreadController

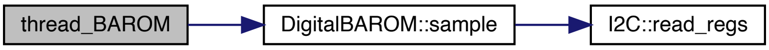
Beep the piezo againHere is the call graph for this function:



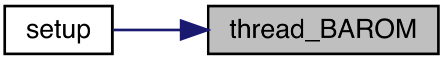
6.8.1.4 thread_BAROM()

```
void thread_BAROM ( )
```

< Sample barometer by calling the Sample functionHere is the call graph for this function:



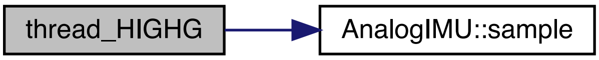
Here is the caller graph for this function:



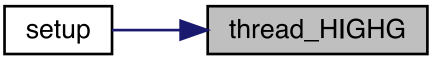
6.8.1.5 thread_HIGHG()

```
void thread_HIGHG ( )
```

< Sample high-g accelerometer by calling the Sample functionHere is the call graph for this function:



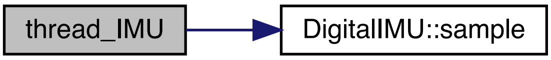
Here is the caller graph for this function:



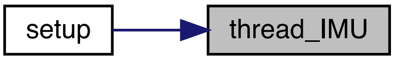
6.8.1.6 thread_IMU()

```
void thread_IMU ( )
```

< Sample the IMU by calling the IMU Sample functionHere is the call graph for this function:



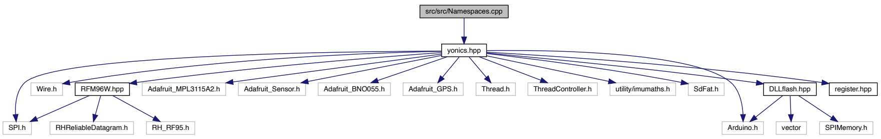
Here is the caller graph for this function:



6.9 src/src/Namespaces.cpp File Reference

```
#include "yonics.hpp"
```

Include dependency graph for Namespaces.cpp:



Namespaces

- INITS
- PROTOTHREADING

  PROTOTHREADING DEFINITIONS.
- I2C

  I2C DRIVERS.

Functions

- bool I2C::write_reg (uint8_t i2c, uint8_t addr, uint8_t val)
- bool I2C::read_regs (uint8_t i2c, uint8_t addr, uint8_t ∗data, uint8_t num)
- bool I2C::read_regs (uint8_t i2c, uint8_t ∗data, uint8_t num)

Variables

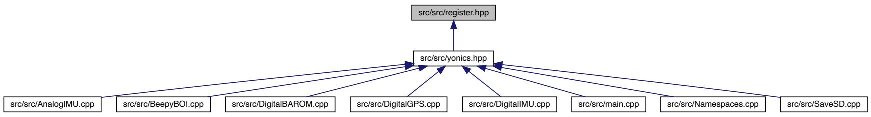
- int INITS::speakerPin = 2
- int INITS::highG_xPin = 33
- int INITS::highG_yPin = 34
- int INITS::highG_zPin = 35
- DigitalIMU INITS::IMU = DigitalIMU(55,0x28)
- DigitalBAROM INITS::BAROM
- AnalogIMU INITS::HIGHG = AnalogIMU(highG_xPin,highG_yPin,highG_zPin,true)
- BeepyBOI INITS::berp = BeepyBOI(speakerPin)
- DLLflash ∗ INITS::flash
- GPSdata INITS::gps_data
- IMUdata INITS::imu_data
- BAROMdata INITS::barom_data
- ACCELdata INITS::accel_data
- int PROTOTHREADING::interval_IMU = 45

  PROTOTHREADING TIME INTERVALS.
- int PROTOTHREADING::interval_BAROM = 2000
- int PROTOTHREADING::interval_ACCEL = 50
- ThreadController PROTOTHREADING::thread_control = ThreadController()

  PROTOTHREADING Declaration and Definitions.
- Thread ∗ PROTOTHREADING::ThreadIMU = new Thread()
- Thread ∗ PROTOTHREADING::ThreadBAROM = new Thread()
- Thread ∗ PROTOTHREADING::ThreadACCEL = new Thread()

6.10 src/src/register.hpp File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define BNO055_MAG_RADIUS_MSB 0x6A
- #define BNO055_MAG_RADIUS_LSB 0x69
- #define BNO055_ACC_RADIUS_MSB 0x68
- #define BNO055_ACC_RADIUS_LSB 0x67
- #define BNO055_GYR_OFFSET_Z_MSB 0x66
- #define BNO055_GYR_OFFSET_Z_LSB 0x65
- #define BNO055_GYR_OFFSET_Y_MSB 0x64
- #define BNO055_GYR_OFFSET_Y_LSB 0x63
- #define BNO055_GYR_OFFSET_X_MSB 0x62
- #define BNO055_GYR_OFFSET_X_LSB 0x61
- #define BNO055_MAG_OFFSET_Z_MSB 0x60
- #define BNO055_MAG_OFFSET_Z_LSB 0x5F
- #define BNO055_MAG_OFFSET_Y_MSB 0x5E
- #define BNO055_MAG_OFFSET_Y_LSB 0x5D
- #define BNO055_MAG_OFFSET_X_MSB 0x5C
- #define BNO055_MAG_OFFSET_X_LSB 0x5B
- #define BNO055_ACC_OFFSET_Z_MSB 0x5A
- #define BNO055_ACC_OFFSET_Z_LSB 0x59
- #define BNO055_ACC_OFFSET_Y_MSB 0x58
- #define BNO055_ACC_OFFSET_Y_LSB 0x57
- #define BNO055_ACC_OFFSET_X_MSB 0x56
- #define BNO055_ACC_OFFSET_X_LSB 0x55
- #define BNO055_AXIS_MAP_SIGN 0x42
- #define BNO055_AXIS_MAP_CONFIG 0x41
- #define BNO055_TEMP_SOURCE 0x40
- #define BNO055_SYS_TRIGGER 0x3F
- #define BNO055_PWR_MODE 0x3E
- #define BNO055_OPR_MODE 0x3D
- #define BNO055_Reserved 0x3C
- #define BNO055_UNIT_SEL 0x3B
- #define BNO055_SYS_ERR 0x3A
- #define BNO055_SYS_STATUS 0x39
- #define BNO055_SYS_CLK_STA 0x38
- #define BNO055_INT_STA 0x37
- #define BNO055_ST_RESULT 0x36
- #define BNO055_CALIB_STAT 0x35
- #define BNO055_TEMP 0x34
- #define BNO055_GRV_Data_Z_M 0x33
- #define BNO055_GRV_Data_Z_L 0x32
- #define BNO055_GRV_Data_Y_M 0x31
- #define BNO055_GRV_Data_Y_L 0x30
- #define BNO055_GRV_Data_X_M 0x2F
- #define BNO055_GRV_Data_X_L 0x2E
- #define BNO055_LIA_Data_Z_MB 0x2D
- #define BNO055_LIA_Data_Z_LS 0x2C
- #define BNO055_LIA_Data_Y_MB 0x2B
- #define BNO055_LIA_Data_Y_LS 0x2A
- #define BNO055_LI A_Data_X_MB 0x29
- #define BNO055_LIA_Data_X_LS 0x28
- #define BNO055_QUA_Data_z_M 0x27
- #define BNO055_QUA_Data_z_LS 0x26
- #define BNO055_QUA_Data_y_M 0x25
- #define BNO055_QUA_Data_y_LS 0x24

- #define BNO055_QUA_Data_x_M 0x23
- #define BNO055_QUA_Data_x_LS 0x22
- #define BNO055_QUA_Data_w_M 0x21
- #define BNO055_QUA_Data_w_L 0x20
- #define BNO055_EUL_Pitch_MSB 0x1F
- #define BNO055_EUL_Pitch_LSB 0x1E
- #define BNO055_EUL_Roll_MSB 0x1D
- #define BNO055_EUL_Roll_LSB 0x1C
- #define BNO055_EUL_Heading_MSB 0x1B
- #define BNO055_EUL_Heading_LSB 0x1A
- #define BNO055_GYR_DATA_Z_MSB 0x19
- #define BNO055_GYR_DATA_Z_LSB 0x18
- #define BNO055_GYR_DATA_Y_MSB 0x17
- #define BNO055_GYR_DATA_Y_LSB 0x16
- #define BNO055_GYR_DATA_X_MSB 0x15
- #define BNO055_GYR_DATA_X_LSB 0x14
- #define BNO055_MAG_DATA_Z_MSB 0x13
- #define BNO055_MAG_DATA_Z_LSB 0x12
- #define BNO055_MAG_DATA_Y_MSB 0x11
- #define BNO055_MAG_DATA_Y_LSB 0x10
- #define BNO055_MAG_DATA_X_MSB 0x0F
- #define BNO055_MAG_DATA_X_LSB 0x0E
- #define BNO055_ACC_DATA_Z_MSB 0x0D
- #define BNO055_ACC_DATA_Z_LSB 0x0C
- #define BNO055_ACC_DATA_Y_MSB 0x0B
- #define BNO055_ACC_DATA_Y_LSB 0x0A
- #define BNO055_ACC_DATA_X_MSB 0x09
- #define BNO055_PWR_ACC_DATA_X_LSB 0x08
- #define BNO055_Page_ID 0x07
- #define BNO055_BL_Rev_ID 0x06
- #define BNO055_SW_REV_ID_MSB 0x05
- #define BNO055_SW_REV_ID_LSB 0x04
- #define BNO055_GYR_ID 0x03
- #define BNO055_MAG_ID 0x02
- #define BNO055_ACC_ID 0x01
- #define BNO055_CHIP_ID 0x00
- #define MPL3115_I2C_ADDR 0x60
- #define MPL3115_STATUS 0x00
- #define MPL3115_OUT_P_MSB 0x01
- #define MPL3115_OUT_P_CSB 0x02
- #define MPL3115_OUT_P_LSB 0x03
- #define MPL3115_OUT_T_MSB 0x04
- #define MPL3115_OUT_T_LSB 0x05
- #define MPL3115_DR_STATUS 0x06
- #define MPL3115_OUT_P_DELTA MSB 0x07
- #define MPL3115_OUT_P_DELTA_CSB 0x08
- #define MPL3115_OUT_P_DELTA_LSB 0x09
- #define MPL3115_OUT_T_DELTA_MSB 0x0A
- #define MPL3115_OUT_T_DELTA_LSB 0x0B
- #define MPL3115_WHO_AM_I 0x0C
- #define MPL3115_F_STATUS 0x0D
- #define MPL3115_F_DATA 0X0E
- #define MPL3115_F_SETUP 0x0F
- #define MPL3115_TIME_DLY 0x10
- #define MPL3115_SYSMOD 0x11

- #define MPL3115_INT_SOURCE 0x12
- #define MPL3115_PT_DATA_CFG 0x13
- #define MPL3115_BAR_IN_MSB 0x14
- #define MPL3115_BAR_IN_LSB 0x15
- #define MPL3115_P_TGT_MSB 0x16
- #define MPL3115_P_TGT_LSB 0x17
- #define MPL3115_T_TGT 0x18
- #define MPL3115_P_WND_MSB 0x19
- #define MPL3115_P_WND_LSB 0x1A
- #define MPL3115_T_WND 0X1B
- #define MPL3115_P_MIN_MSB 0x1C
- #define MPL3115_P_MIN_CSB 0x1D
- #define MPL3115_P_MIN_LSB 0x1E
- #define MPL3115_T_MIN_MSB 0x1F
- #define MPL3115_T_MIN_LSB 0x20
- #define MPL3115_P_MAX_MSB 0x21
- #define MPL3115_P_MAX_CSB 0x22
- #define MPL3115_P_MAX_LSB 0x23
- #define MPL3115_T_MAX_MSB 0X24
- #define MPL3115_T_MAX_LSB 0x25
- #define MPL3115_CTRL_REG1 0x26
- #define MPL3115_CTRL_REG2 0x27
- #define MPL3115_CTRL_REG3 0x28
- #define MPL3115_CTRL_REG4 0x29
- #define MPL3115_CTRL_REG5 0x2A
- #define MPL3115_OFF_P 0x2B
- #define MPL3115_OFF_T 0x2C
- #define MPL3115_OFF_H 0x2D

6.10.1 Macro Definition Documentation

6.10.1.1 BNO055_ACC_DATA_X_MSB

```
#define BNO055_ACC_DATA_X_MSB 0x09
```

6.10.1.2 BNO055_ACC_DATA_Y_LSB

```
#define BNO055_ACC_DATA_Y_LSB 0x0A
```

6.10.1.3 BNO055_ACC_DATA_Y_MSB

```
#define BNO055_ACC_DATA_Y_MSB 0x0B
```

6.10.1.4 BNO055_ACC_DATA_Z_LSB

```
#define BNO055_ACC_DATA_Z_LSB 0x0C
```

6.10.1.5 BNO055_ACC_DATA_Z_MSB

```
#define BNO055_ACC_DATA_Z_MSB 0x0D
```

6.10.1.6 BNO055_ACC_ID

```
#define BNO055_ACC_ID 0x01
```

6.10.1.7 BNO055_ACC_OFFSET_X_LSB

```
#define BNO055_ACC_OFFSET_X_LSB 0x55
```

6.10.1.8 BNO055_ACC_OFFSET_X_MSB

```
#define BNO055_ACC_OFFSET_X_MSB 0x56
```

6.10.1.9 BNO055_ACC_OFFSET_Y_LSB

```
#define BNO055_ACC_OFFSET_Y_LSB 0x57
```

6.10.1.10 BNO055_ACC_OFFSET_Y_MSB

```
#define BNO055_ACC_OFFSET_Y_MSB 0x58
```

6.10.1.11 BNO055_ACC_OFFSET_Z_LSB

```
#define BNO055_ACC_OFFSET_Z_LSB 0x59
```

6.10.1.12 BNO055_ACC_OFFSET_Z_MSB

```
#define BNO055_ACC_OFFSET_Z_MSB 0x5A
```

6.10.1.13 BNO055_ACC_RADIUS_LSB

```
#define BNO055_ACC_RADIUS_LSB 0x67
```

6.10.1.14 BNO055_ACC_RADIUS_MSB

```
#define BNO055_ACC_RADIUS_MSB 0x68
```

6.10.1.15 BNO055_AXIS_MAP_CONFIG

```
#define BNO055_AXIS_MAP_CONFIG 0x41
```

6.10.1.16 BNO055_AXIS_MAP_SIGN

```
#define BNO055_AXIS_MAP_SIGN 0x42
```

6.10.1.17 BNO055_BL_Rev_ID

```
#define BNO055_BL_Rev_ID 0x06
```

6.10.1.18 BNO055_CALIB_STAT

```
#define BNO055_CALIB_STAT 0x35
```

6.10.1.19 BNO055_CHIP_ID

```
#define BNO055_CHIP_ID 0x00
```

6.10.1.20 BNO055_EUL_Heading_LSB

```
#define BNO055_EUL_Heading_LSB 0x1A
```

6.10.1.21 BNO055_EUL_Heading_MSB

```
#define BNO055_EUL_Heading_MSB 0x1B
```

6.10.1.22 BNO055_EUL_Pitch_LSB

```
#define BNO055_EUL_Pitch_LSB 0x1E
```

6.10.1.23 BNO055_EUL_Pitch_MSB

```
#define BNO055_EUL_Pitch_MSB 0x1F
```

6.10.1.24 BNO055_EUL_Roll_LSB

```
#define BNO055_EUL_Roll_LSB 0x1C
```

6.10.1.25 BNO055_EUL_Roll_MSB

```
#define BNO055_EUL_Roll_MSB 0x1D
```

6.10.1.26 BNO055_GRV_Data_X_L

```
#define BNO055_GRV_Data_X_L 0x2E
```

6.10.1.27 BNO055_GRV_Data_X_M

```
#define BNO055_GRV_Data_X_M 0x2F
```

6.10.1.28 BNO055_GRV_Data_Y_L

```
#define BNO055_GRV_Data_Y_L 0x30
```

6.10.1.29 BNO055_GRV_Data_Y_M

```
#define BNO055_GRV_Data_Y_M 0x31
```

6.10.1.30 BNO055_GRV_Data_Z_L

```
#define BNO055_GRV_Data_Z_L 0x32
```

6.10.1.31 BNO055_GRV_Data_Z_M

```
#define BNO055_GRV_Data_Z_M 0x33
```

6.10.1.32 BNO055_GYR_DATA_X_LSB

```
#define BNO055_GYR_DATA_X_LSB 0x14
```

6.10.1.33 BNO055_GYR_DATA_X_MSB

```
#define BNO055_GYR_DATA_X_MSB 0x15
```

6.10.1.34 BNO055_GYR_DATA_Y_LSB

```
#define BNO055_GYR_DATA_Y_LSB 0x16
```

6.10.1.35 BNO055_GYR_DATA_Y_MSB

```
#define BNO055_GYR_DATA_Y_MSB 0x17
```

6.10.1.36 BNO055_GYR_DATA_Z_LSB

```
#define BNO055_GYR_DATA_Z_LSB 0x18
```

6.10.1.37 BNO055_GYR_DATA_Z_MSB

```
#define BNO055_GYR_DATA_Z_MSB 0x19
```

6.10.1.38 BNO055_GYR_ID

```
#define BNO055_GYR_ID 0x03
```

6.10.1.39 BNO055_GYR_OFFSET_X_LSB

```
#define BNO055_GYR_OFFSET_X_LSB 0x61
```

6.10.1.40 BNO055_GYR_OFFSET_X_MSB

```
#define BNO055_GYR_OFFSET_X_MSB 0x62
```

6.10.1.41 BNO055_GYR_OFFSET_Y_LSB

```
#define BNO055_GYR_OFFSET_Y_LSB 0x63
```

6.10.1.42 BNO055_GYR_OFFSET_Y_MSB

```
#define BNO055_GYR_OFFSET_Y_MSB 0x64
```

6.10.1.43 BNO055_GYR_OFFSET_Z_LSB

```
#define BNO055_GYR_OFFSET_Z_LSB 0x65
```

6.10.1.44 BNO055_GYR_OFFSET_Z_MSB

```
#define BNO055_GYR_OFFSET_Z_MSB 0x66
```

6.10.1.45 BNO055_INT_STA

```
#define BNO055_INT_STA 0x37
```

6.10.1.46 BNO055_LI

```
#define BNO055_LI A_Data_X_MB 0x29
```

6.10.1.47 BNO055_LIA_Data_X_LS

```
#define BNO055_LIA_Data_X_LS 0x28
```

6.10.1.48 BNO055_LIA_Data_Y_LS

```
#define BNO055_LIA_Data_Y_LS 0x2A
```

6.10.1.49 BNO055_LIA_Data_Y_MB

```
#define BNO055_LIA_Data_Y_MB 0x2B
```

6.10.1.50 BNO055_LIA_Data_Z_LS

```
#define BNO055_LIA_Data_Z_LS 0x2C
```

6.10.1.51 BNO055_LIA_Data_Z_MB

```
#define BNO055_LIA_Data_Z_MB 0x2D
```

6.10.1.52 BNO055_MAG_DATA_X_LSB

```
#define BNO055_MAG_DATA_X_LSB 0x0E
```

6.10.1.53 BNO055_MAG_DATA_X_MSB

```
#define BNO055_MAG_DATA_X_MSB 0x0F
```

6.10.1.54 BNO055_MAG_DATA_Y_LSB

```
#define BNO055_MAG_DATA_Y_LSB 0x10
```

6.10.1.55 BNO055_MAG_DATA_Y_MSB

```
#define BNO055_MAG_DATA_Y_MSB 0x11
```

6.10.1.56 BNO055_MAG_DATA_Z_LSB

```
#define BNO055_MAG_DATA_Z_LSB 0x12
```

6.10.1.57 BNO055_MAG_DATA_Z_MSB

```
#define BNO055_MAG_DATA_Z_MSB 0x13
```

6.10.1.58 BNO055_MAG_ID

```
#define BNO055_MAG_ID 0x02
```

6.10.1.59 BNO055_MAG_OFFSET_X_LSB

```
#define BNO055_MAG_OFFSET_X_LSB 0x5B
```

6.10.1.60 BNO055_MAG_OFFSET_X_MSB

```
#define BNO055_MAG_OFFSET_X_MSB 0x5C
```

6.10.1.61 BNO055_MAG_OFFSET_Y_LSB

```
#define BNO055_MAG_OFFSET_Y_LSB 0x5D
```

6.10.1.62 BNO055_MAG_OFFSET_Y_MSB

```
#define BNO055_MAG_OFFSET_Y_MSB 0x5E
```

6.10.1.63 BNO055_MAG_OFFSET_Z_LSB

```
#define BNO055_MAG_OFFSET_Z_LSB 0x5F
```

6.10.1.64 BNO055_MAG_OFFSET_Z_MSB

```
#define BNO055_MAG_OFFSET_Z_MSB 0x60
```

6.10.1.65 BNO055_MAG_RADIUS_LSB

```
#define BNO055_MAG_RADIUS_LSB 0x69
```

6.10.1.66 BNO055_MAG_RADIUS_MSB

```
#define BNO055_MAG_RADIUS_MSB 0x6A
```

6.10.1.67 BNO055_OPR_MODE

```
#define BNO055_OPR_MODE 0x3D
```

6.10.1.68 BNO055_Page_ID

```
#define BNO055_Page_ID 0x07
```

6.10.1.69 BNO055_PWR_ACC_DATA_X_LSB

```
#define BNO055_PWR_ACC_DATA_X_LSB 0x08
```

6.10.1.70 BNO055_PWR_MODE

```
#define BNO055_PWR_MODE 0x3E
```

6.10.1.71 BNO055_QUA_Data_w_L

```
#define BNO055_QUA_Data_w_L 0x20
```

6.10.1.72 BNO055_QUA_Data_w_M

```
#define BNO055_QUA_Data_w_M 0x21
```

6.10.1.73 BNO055_QUA_Data_x_LS

```
#define BNO055_QUA_Data_x_LS 0x22
```

6.10.1.74 BNO055_QUA_Data_x_M

```
#define BNO055_QUA_Data_x_M 0x23
```

6.10.1.75 BNO055_QUA_Data_y_LS

```
#define BNO055_QUA_Data_y_LS 0x24
```

6.10.1.76 BNO055_QUA_Data_y_M

```
#define BNO055_QUA_Data_y_M 0x25
```

6.10.1.77 BNO055_QUA_Data_z_LS

```
#define BNO055_QUA_Data_z_LS 0x26
```

6.10.1.78 BNO055_QUA_Data_z_M

```
#define BNO055_QUA_Data_z_M 0x27
```

6.10.1.79 BNO055_Reserved

```
#define BNO055_Reserved 0x3C
```

6.10.1.80 BNO055_ST_RESULT

```
#define BNO055_ST_RESULT 0x36
```

6.10.1.81 BNO055_SW_REV_ID_LSB

```
#define BNO055_SW_REV_ID_LSB 0x04
```

6.10.1.82 BNO055_SW_REV_ID_MSB

```
#define BNO055_SW_REV_ID_MSB 0x05
```

6.10.1.83 BNO055_SYS_CLK_STA

```
#define BNO055_SYS_CLK_STA 0x38
```

6.10.1.84 BNO055_SYS_ERR

```
#define BNO055_SYS_ERR 0x3A
```

6.10.1.85 BNO055_SYS_STATUS

```
#define BNO055_SYS_STATUS 0x39
```

6.10.1.86 BNO055_SYS_TRIGGER

```
#define BNO055_SYS_TRIGGER 0x3F
```

6.10.1.87 BNO055_TEMP

```
#define BNO055_TEMP 0x34
```

6.10.1.88 BNO055_TEMP_SOURCE

```
#define BNO055_TEMP_SOURCE 0x40
```

6.10.1.89 BNO055_UNIT_SEL

```
#define BNO055_UNIT_SEL 0x3B
```

6.10.1.90 MPL3115_BAR_IN_LSB

```
#define MPL3115_BAR_IN_LSB 0x15
```

6.10.1.91 MPL3115_BAR_IN_MSB

```
#define MPL3115_BAR_IN_MSB 0x14
```

6.10.1.92 MPL3115_CTRL_REG1

```
#define MPL3115_CTRL_REG1 0x26
```

6.10.1.93 MPL3115_CTRL_REG2

```
#define MPL3115_CTRL_REG2 0x27
```

6.10.1.94 MPL3115_CTRL_REG3

```
#define MPL3115_CTRL_REG3 0x28
```

6.10.1.95 MPL3115_CTRL_REG4

```
#define MPL3115_CTRL_REG4 0x29
```

6.10.1.96 MPL3115_CTRL_REG5

```
#define MPL3115_CTRL_REG5 0x2A
```

6.10.1.97 MPL3115_DR_STATUS

```
#define MPL3115_DR_STATUS 0x06
```

6.10.1.98 MPL3115_F_DATA

```
#define MPL3115_F_DATA 0X0E
```

6.10.1.99 MPL3115_F_SETUP

```
#define MPL3115_F_SETUP 0x0F
```

6.10.1.100 MPL3115_F_STATUS

```
#define MPL3115_F_STATUS 0x0D
```

6.10.1.101 MPL3115_I2C_ADDR

```
#define MPL3115_I2C_ADDR 0x60
```

6.10.1.102 MPL3115_INT_SOURCE

```
#define MPL3115_INT_SOURCE 0x12
```

6.10.1.103 MPL3115_OFF_H

```
#define MPL3115_OFF_H 0x2D
```

6.10.1.104 MPL3115_OFF_P

```
#define MPL3115_OFF_P 0x2B
```

6.10.1.105 MPL3115_OFF_T

```
#define MPL3115_OFF_T 0x2C
```

6.10.1.106 MPL3115_OUT_P_CSB

```
#define MPL3115_OUT_P_CSB 0x02
```

6.10.1.107 MPL3115_OUT_P_DELTA

```
#define MPL3115_OUT_P_DELTA MSB 0x07
```

6.10.1.108 MPL3115_OUT_P_DELTA_CSB

```
#define MPL3115_OUT_P_DELTA_CSB 0x08
```

6.10.1.109 MPL3115_OUT_P_DELTA_LSB

```
#define MPL3115_OUT_P_DELTA_LSB 0x09
```

6.10.1.110 MPL3115_OUT_P_LSB

```
#define MPL3115_OUT_P_LSB 0x03
```

6.10.1.111 MPL3115_OUT_P_MSB

```
#define MPL3115_OUT_P_MSB 0x01
```

6.10.1.112 MPL3115_OUT_T_DELTA_LSB

```
#define MPL3115_OUT_T_DELTA_LSB 0x0B
```

6.10.1.113 MPL3115_OUT_T_DELTA_MSB

```
#define MPL3115_OUT_T_DELTA_MSB 0x0A
```

6.10.1.114 MPL3115_OUT_T_LSB

```
#define MPL3115_OUT_T_LSB 0x05
```

6.10.1.115 MPL3115_OUT_T_MSB

```
#define MPL3115_OUT_T_MSB 0x04
```

6.10.1.116 MPL3115_P_MAX_CSB

```
#define MPL3115_P_MAX_CSB 0x22
```

6.10.1.117 MPL3115_P_MAX_LSB

```
#define MPL3115_P_MAX_LSB 0x23
```

6.10.1.118 MPL3115_P_MAX_MSB

```
#define MPL3115_P_MAX_MSB 0x21
```

6.10.1.119 MPL3115_P_MIN_CSB

```
#define MPL3115_P_MIN_CSB 0x1D
```

6.10.1.120 MPL3115_P_MIN_LSB

```
#define MPL3115_P_MIN_LSB 0x1E
```

6.10.1.121 MPL3115_P_MIN_MSB

```
#define MPL3115_P_MIN_MSB 0x1C
```

6.10.1.122 MPL3115_P_TGT_LSB

```
#define MPL3115_P_TGT_LSB 0x17
```

6.10.1.123 MPL3115_P_TGT_MSB

```
#define MPL3115_P_TGT_MSB 0x16
```

6.10.1.124 MPL3115_P_WND_LSB

```
#define MPL3115_P_WND_LSB 0x1A
```

6.10.1.125 MPL3115_P_WND_MSB

```
#define MPL3115_P_WND_MSB 0x19
```

6.10.1.126 MPL3115_PT_DATA_CFG

```
#define MPL3115_PT_DATA_CFG 0x13
```

6.10.1.127 MPL3115_STATUS

```
#define MPL3115_STATUS 0x00
```

6.10.1.128 MPL3115_SYSMOD

```
#define MPL3115_SYSMOD 0x11
```

6.10.1.129 MPL3115_T_MAX_LSB

```
#define MPL3115_T_MAX_LSB 0x25
```

6.10.1.130 MPL3115_T_MAX_MSB

```
#define MPL3115_T_MAX_MSB 0X24
```

6.10.1.131 MPL3115_T_MIN_LSB

```
#define MPL3115_T_MIN_LSB 0x20
```

6.10.1.132 MPL3115_T_MIN_MSB

```
#define MPL3115_T_MIN_MSB 0x1F
```

6.10.1.133 MPL3115_T_TGT

```
#define MPL3115_T_TGT 0x18
```

6.10.1.134 MPL3115_T_WND

```
#define MPL3115_T_WND 0X1B
```

6.10.1.135 MPL3115_TIME_DLY

```
#define MPL3115_TIME_DLY 0x10
```

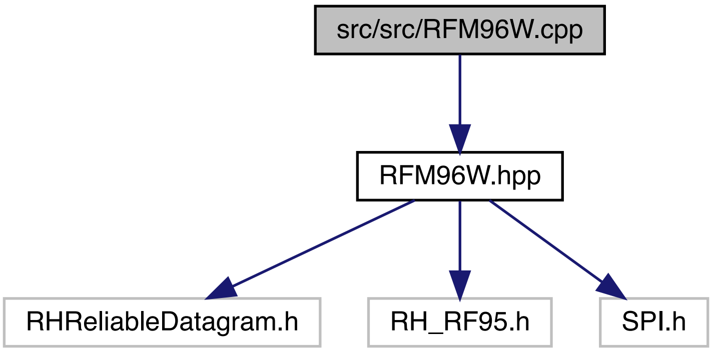
6.10.1.136 MPL3115_WHO_AM_I

```
#define MPL3115_WHO_AM_I 0x0C
```

6.11 src/src/RFM96W.cpp File Reference

```
#include "RFM96W.hpp"
```

Include dependency graph for RFM96W.cpp:



Variables

- uint8_t ServerReturnMessage [ ] = "RECVD"
- uint8_t server_buf [RH_RF95_MAX_MESSAGE_LEN]
- uint8_t server_len = sizeof(server_buf)
- uint8_t server_from
- uint8_t client_buf [RH_RF95_MAX_MESSAGE_LEN]
- uint8_t client_len = sizeof(client_buf)
- uint8_t client_from

6.11.1 Variable Documentation

6.11.1.1 client_buf

```
uint8_t client_buf[RH_RF95_MAX_MESSAGE_LEN]
```

6.11.1.2 client_from

```
uint8_t client_from
```

6.11.1.3 client_len

```
uint8_t client_len = sizeof(client_buf)
```

6.11.1.4 server_buf

```
uint8_t server_buf[RH_RF95_MAX_MESSAGE_LEN]
```

6.11.1.5 server_from

```
uint8_t server_from
```

6.11.1.6 server_len

```
uint8_t server_len = sizeof(server_buf)
```

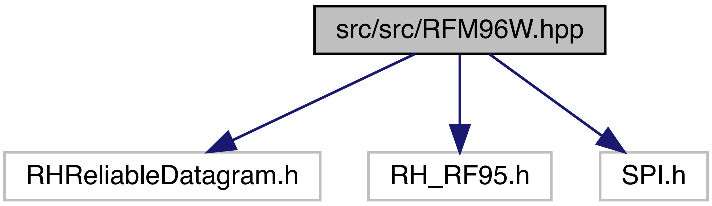
6.11.1.7 ServerReturnMessage

```
uint8_t ServerReturnMessage[ ] = "RECVD"
```

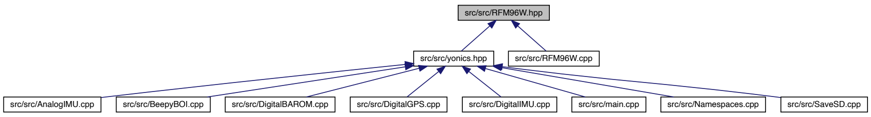
6.12 src/src/RFM96W.hpp File Reference

```
#include <RHReliableDatagram.h>
#include <RH_RF95.h>
#include <SPI.h>
```

Include dependency graph for RFM96W.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class RFM96W_Server
- class RFM96W_Client

Macros

- #define CLIENT_ADDRESS 1
- #define SERVER_ADDRESS 2

6.12.1 Macro Definition Documentation

6.12.1.1 CLIENT_ADDRESS

```
#define CLIENT_ADDRESS 1
```

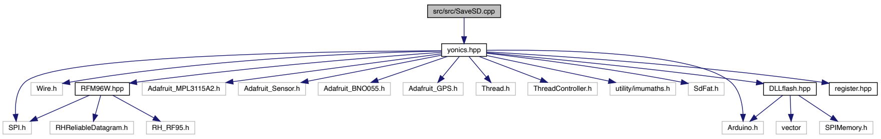
6.12.1.2 SERVER_ADDRESS

```
#define SERVER_ADDRESS 2
```

6.13 src/src/SaveSD.cpp File Reference

```
#include "yonics.hpp"
```

Include dependency graph for SaveSD.cpp:



6.14 src/src/StructDefs.hpp File Reference

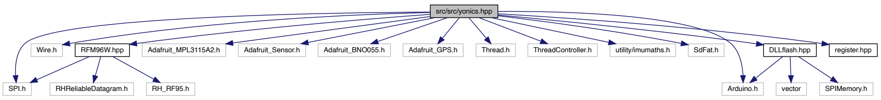
Classes

- struct GPSdata
- struct ACCELdata
- struct IMUdata
- struct BAROMdata

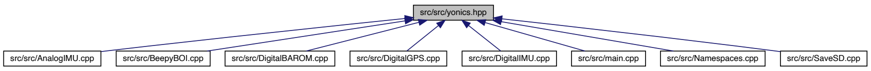
6.15 src/src/yonics.hpp File Reference

```
#include <Arduino.h>
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_MPL3115A2.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BNO055.h>
#include <Adafruit_GPS.h>
#include "Thread.h"
#include <ThreadController.h>
#include <utility/imumaths.h>
#include <SdFat.h>
#include "RFM96W.hpp"
#include "DLLflash.hpp"
#include "register.hpp"
```

Include dependency graph for yonics.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- struct GPSdata
- struct ACCELdata
- struct IMUdata
- struct BAROMdata
- class SaveSD
- class AnalogIMU
- class DigitalIMU
- class DigitalBAROM
- class DigitalGPS
- class BeepyBOI

Namespaces

- I2C

  I2C DRIVERS.
- INITS
- PROTOTHREADING

  PROTOTHREADING DEFINITIONS.

Macros

- #define GPSSerial Serial3

  HEADER FILES.

Functions

- bool I2C::write_reg (uint8_t i2c, uint8_t addr, uint8_t val)
- bool I2C::read_regs (uint8_t i2c, uint8_t addr, uint8_t ∗data, uint8_t num)
- bool I2C::read_regs (uint8_t i2c, uint8_t ∗data, uint8_t num)

6.15.1 Macro Definition Documentation

6.15.1.1 GPSSerial

```
#define GPSSerial Serial3
```

HEADER FILES.

All the main header files that the project uses are linked here. Libraries, Sensors, Etc... MACROS

Some MACROS that need to be defined beforehand Define the GPS hardware Serial Port