University of Colorado - Boulder

Sounding Rocket Lab Avionics

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

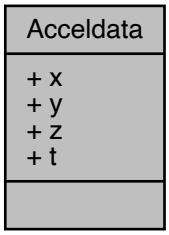
Chapter 3

Class Documentation

3.1 Acceldata Struct Reference

`#include <yonics.hpp>`

Collaboration diagram for Acceldata:



Public Attributes

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

3.1.1 Detailed Description

Definition at line 24 of file yonics.hpp.

3.1.2 Member Data Documentation

3.1.2.1 t

```
uint32_t Acceldata::t
```

Definition at line 28 of file yonics.hpp.

3.1.2.2 x

```
float Acceldata::x
```

Definition at line 25 of file yonics.hpp.

3.1.2.3 y

```
float Acceldata::y
```

Definition at line 26 of file yonics.hpp.

3.1.2.4 z

```
float Acceldata::z
```

Definition at line 27 of file yonics.hpp.

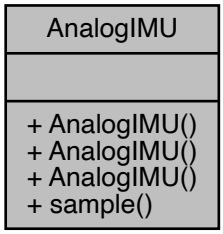
The documentation for this struct was generated from the following file:

- src/src/yonics.hpp

3.2 AnalogIMU Class Reference

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

Collaboration diagram for AnalogIMU:



Public Member Functions

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

3.2.1 Detailed Description

Definition at line 132 of file yonics.hpp.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 AnalogIMU() [1/3]

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

Definition at line 3 of file AnalogIMU.cpp.

3.2.2.2 AnalogIMU() [2/3]

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

Definition at line 12 of file AnalogIMU.cpp.

3.2.2.3 AnalogIMU() [3/3]

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

Definition at line 21 of file AnalogIMU.cpp.

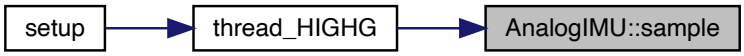
3.2.3 Member Function Documentation

3.2.3.1 sample()

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

Definition at line 42 of file AnalogIMU.cpp.

Here is the caller graph for this function:



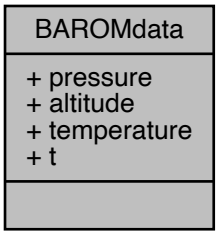
The documentation for this class was generated from the following files:

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

3.3 BAROMdata Struct Reference

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

Collaboration diagram for BAROMdata:



Public Attributes

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

3.3.1 Detailed Description

Definition at line 42 of file yonics.hpp.

3.3.2 Member Data Documentation

3.3.2.1 altitude

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

Definition at line 44 of file yonics.hpp.

3.3.2.2 pressure

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

Definition at line 43 of file yonics.hpp.

3.3.2.3 t

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

Definition at line 46 of file yonics.hpp.

3.3.2.4 temperature

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

Definition at line 45 of file yonics.hpp.

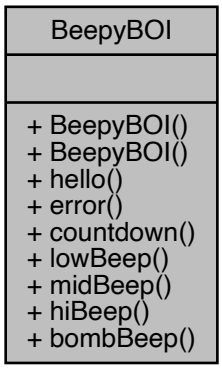
The documentation for this struct was generated from the following file:

- src/src/yonics.hpp

3.4 BeepyBOI Class Reference

`#include <yonics.hpp>`

Collaboration diagram for BeepyBOI:



Public Member Functions

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

3.4.1 Detailed Description

Definition at line 190 of file yonics.hpp.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 BeepyBOI() [1/2]

`BeepyBOI::BeepyBOI ( )`

Definition at line 3 of file BeepyBOI.cpp.

3.4.2.2 BeepyBOI() [2/2]

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

Definition at line 7 of file BeepyBOI.cpp.

3.4.3 Member Function Documentation

3.4.3.1 bombBeep()

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

Definition at line 43 of file BeepyBOI.cpp.

Here is the caller graph for this function:

3.4.3.2 countdown()

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

Definition at line 21 of file BeepyBOI.cpp.

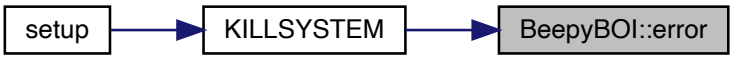
Here is the caller graph for this function:

3.4.3.3 error()

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

Definition at line 16 of file BeepyBOI.cpp.

Here is the caller graph for this function:

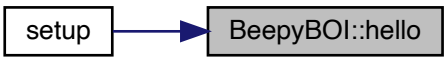


3.4.3.4 hello()

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

Definition at line 11 of file BeepyBOI.cpp.

Here is the caller graph for this function:



3.4.3.5 hiBeep()

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

Definition at line 38 of file BeepyBOI.cpp.

Here is the caller graph for this function:

3.4.3.6 lowBeep()

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

Definition at line 28 of file BeepyBOI.cpp.

Here is the caller graph for this function:

3.4.3.7 midBeep()

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

Definition at line 33 of file BeepyBOI.cpp.

Here is the caller graph for this function:

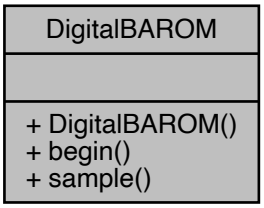
The documentation for this class was generated from the following files:

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

3.5 DigitalBAROM Class Reference

`#include <yonics.hpp>`

Collaboration diagram for DigitalBAROM:



Public Member Functions

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

3.5.1 Detailed Description

Definition at line 164 of file yonics.hpp.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 DigitalBAROM()

`DigitalBAROM::DigitalBAROM ( )`

Definition at line 3 of file DigitalBAROM.cpp.

3.5.3 Member Function Documentation

3.5.3.1 begin()

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

Definition at line 5 of file DigitalBAROM.cpp.

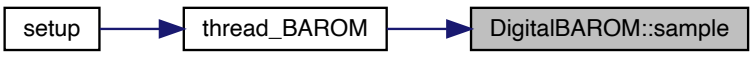
Here is the caller graph for this function:

3.5.3.2 sample()

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

Definition at line 9 of file DigitalBAROM.cpp.

Here is the caller graph for this function:



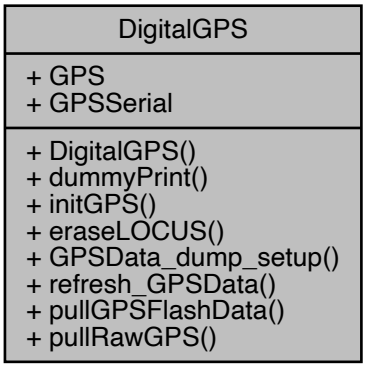
The documentation for this class was generated from the following files:

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

3.6 DigitalGPS Class Reference

`#include <yonics.hpp>`

Collaboration diagram for DigitalGPS:



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

3.6.1 Detailed Description

Definition at line 173 of file yonics.hpp.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 DigitalGPS()

```
DigitalGPS::DigitalGPS (
            HardwareSerial * ser )
```

Definition at line 3 of file DigitalGPS.cpp.

Here is the call graph for this function:

3.6.3 Member Function Documentation

3.6.3.1 dummyPrint()

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

Definition at line 47 of file DigitalGPS.cpp.

3.6.3.2 eraseLOCUS()

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

Definition at line 27 of file DigitalGPS.cpp.

Here is the caller graph for this function:

3.6.3.3 GPSData_dump_setup()

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

Definition at line 33 of file DigitalGPS.cpp.

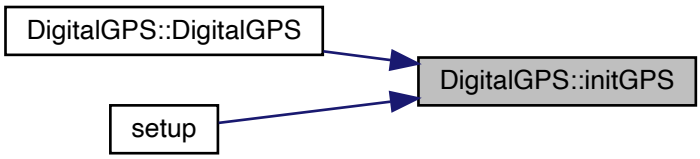
Here is the caller graph for this function:

3.6.3.4 initGPS()

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

Definition at line 14 of file DigitalGPS.cpp.

Here is the caller graph for this function:



3.6.3.5 pullGPSFlashData()

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

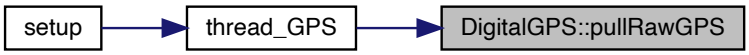
Definition at line 53 of file DigitalGPS.cpp.

3.6.3.6 pullRawGPS()

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

Definition at line 83 of file DigitalGPS.cpp.

Here is the caller graph for this function:

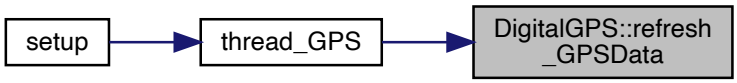


3.6.3.7 refresh_GPSData()

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

Definition at line 62 of file DigitalGPS.cpp.

Here is the caller graph for this function:



3.6.4 Member Data Documentation

3.6.4.1 GPS

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

Definition at line 176 of file yonics.hpp.

3.6.4.2 GPSSerial

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

Definition at line 177 of file yonics.hpp.

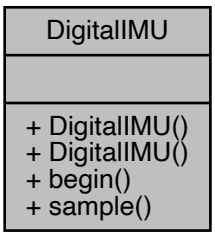
The documentation for this class was generated from the following files:

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

3.7 DigitalIMU Class Reference

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

Collaboration diagram for DigitalIMU:



Public Member Functions

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

3.7.1 Detailed Description

Definition at line 150 of file yonics.hpp.

3.7.2 Constructor & Destructor Documentation

3.7.2.1 DigitalIMU() [1/2]

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

Definition at line 3 of file DigitalIMU.cpp.

3.7.2.2 DigitalIMU() [2/2]

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

Definition at line 7 of file DigitalIMU.cpp.

3.7.3 Member Function Documentation

3.7.3.1 begin()

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

Definition at line 11 of file DigitalIMU.cpp.

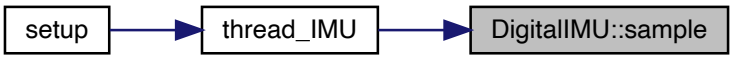
Here is the caller graph for this function:

3.7.3.2 sample()

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

Definition at line 21 of file DigitalIMU.cpp.

Here is the caller graph for this function:



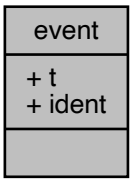
The documentation for this class was generated from the following files:

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

3.8 event Struct Reference

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

Collaboration diagram for event:



Public Attributes

- uint32_t t
- char ident

3.8.1 Detailed Description

Definition at line 57 of file yonics.hpp.

3.8.2 Member Data Documentation

3.8.2.1 ident

```
char event::ident
```

Definition at line 59 of file yonics.hpp.

3.8.2.2 t

```
uint32_t event::t
```

Definition at line 58 of file yonics.hpp.

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

- src/src/yonics.hpp

3.9 FlashOp Class Reference

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

Collaboration diagram for FlashOp:

Public Member Functions

- FlashOp ()
- FlashOp (SPIFlash ∗flash)
- bool beginRead ()
- bool beginWrite ()
- int addType (int size, int interval, void ∗data)
- void addWP (int pin)
- bool addSample (int ident)
- bool addEvent (uint32_t t, char ident)
- bool getType (int ident, int ∗size)
- bool getSample (int ident, int sample, void ∗data)
- bool getEvent (int index, uint32_t ∗t, char ∗ident)
- bool stopReading ()

3.9.1 Detailed Description

Definition at line 62 of file yonics.hpp.

3.9.2 Constructor & Destructor Documentation

3.9.2.1 FlashOp() [1/2]

```
FlashOp::FlashOp ( )
```

Definition at line 3 of file FlashOp.cpp.

3.9.2.2 FlashOp() [2/2]

```
FlashOp::FlashOp (
            SPIFlash * flash )
```

Definition at line 7 of file FlashOp.cpp.

3.9.3 Member Function Documentation

3.9.3.1 addEvent()

```
bool FlashOp::addEvent (
            uint32_t t,
            char ident )
```

Definition at line 143 of file FlashOp.cpp.

3.9.3.2 addSample()

```
bool FlashOp::addSample (
            int ident )
```

Definition at line 121 of file FlashOp.cpp.

3.9.3.3 addType()

```
int FlashOp::addType (
            int size,
            int interval,
            void * data )
```

Definition at line 92 of file FlashOp.cpp.

3.9.3.4 addWP()

```
void FlashOp::addWP (
            int pin )
```

Definition at line 196 of file FlashOp.cpp.

3.9.3.5 beginRead()

```
bool FlashOp::beginRead ( )
```

Definition at line 15 of file FlashOp.cpp.

Here is the caller graph for this function:

3.9.3.6 beginWrite()

```
bool FlashOp::beginWrite ( )
```

Definition at line 35 of file FlashOp.cpp.

3.9.3.7 getEvent()

```
bool FlashOp::getEvent (
            int index,
            uint32_t * t,
            char * ident )
```

Definition at line 183 of file FlashOp.cpp.

3.9.3.8 getSample()

```
bool FlashOp::getSample (
            int ident,
            int sample,
            void * data )
```

Definition at line 169 of file FlashOp.cpp.

3.9.3.9 getType()

```
bool FlashOp::getType (
            int ident,
            int * size )
```

Definition at line 162 of file FlashOp.cpp.

3.9.3.10 stopReading()

```
bool FlashOp::stopReading ( )
```

Definition at line 73 of file FlashOp.cpp.

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

- src/src/yonics.hpp
- src/src/FlashOp.cpp

3.10 GPSdata Struct Reference

`#include <yonics.hpp>`

Collaboration diagram for GPSdata:

Public Attributes

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

3.10.1 Detailed Description

Definition at line 15 of file yonics.hpp.

3.10.2 Member Data Documentation

3.10.2.1 altitude

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

Definition at line 18 of file yonics.hpp.

3.10.2.2 angle

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

Definition at line 20 of file yonics.hpp.

3.10.2.3 lat

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

Definition at line 16 of file yonics.hpp.

3.10.2.4 lon

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

Definition at line 17 of file yonics.hpp.

3.10.2.5 sat_num

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

Definition at line 21 of file yonics.hpp.

3.10.2.6 speed

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

Definition at line 19 of file yonics.hpp.

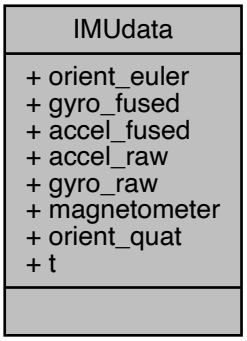
The documentation for this struct was generated from the following file:

- src/src/yonics.hpp

3.11 IMUdata Struct Reference

`#include <yonics.hpp>`

Collaboration diagram for IMUdata:



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

3.11.1 Detailed Description

Definition at line 31 of file yonics.hpp.

3.11.2 Member Data Documentation

3.11.2.1 accel_fused

```
double IMUdata::accel_fused[3] = {0,0,0}
```

Definition at line 34 of file yonics.hpp.

3.11.2.2 accel_raw

```
double IMUdata::accel_raw[3] = {0,0,0}
```

Definition at line 35 of file yonics.hpp.

3.11.2.3 gyro_fused

```
double IMUdata::gyro_fused[3] = {0,0,0}
```

Definition at line 33 of file yonics.hpp.

3.11.2.4 gyro_raw

```
double IMUdata::gyro_raw[3] = {0,0,0}
```

Definition at line 36 of file yonics.hpp.

3.11.2.5 magnetometer

```
double IMUdata::magnetometer[3] = {0,0,0}
```

Definition at line 37 of file yonics.hpp.

3.11.2.6 orient_euler

```
double IMUdata::orient_euler[3] = {0,0,0}
```

Definition at line 32 of file yonics.hpp.

3.11.2.7 orient_quat

```
double IMUdata::orient_quat[4] = {0,0,0,0}
```

Definition at line 38 of file yonics.hpp.

3.11.2.8 t

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

Definition at line 39 of file yonics.hpp.

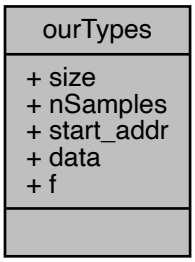
The documentation for this struct was generated from the following file:

- src/src/yonics.hpp

3.12 ourTypes Struct Reference

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

Collaboration diagram for ourTypes:



Public Attributes

- int size = 0
- int nSamples = 0
- uint32_t start_addr = 0
- void * data = NULL
- float f = 0

3.12.1 Detailed Description

Definition at line 49 of file yonics.hpp.

3.12.2 Member Data Documentation

3.12.2.1 data

```
void* ourTypes::data = NULL
```

Definition at line 53 of file yonics.hpp.

3.12.2.2 f

```
float ourTypes::f = 0
```

Definition at line 54 of file yonics.hpp.

3.12.2.3 nSamples

```
int ourTypes::nSamples = 0
```

Definition at line 51 of file yonics.hpp.

3.12.2.4 size

```
int ourTypes::size = 0
```

Definition at line 50 of file yonics.hpp.

3.12.2.5 start_addr

```
uint32_t ourTypes::start_addr = 0
```

Definition at line 52 of file yonics.hpp.

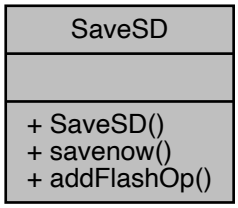
The documentation for this struct was generated from the following file:

- src/src/yonics.hpp

3.13 SaveSD Class Reference

`#include <yonics.hpp>`

Collaboration diagram for SaveSD:



Public Member Functions

- SaveSD ()
- bool savenow ()
- bool addFlashOp (FlashOp ∗flash)

3.13.1 Detailed Description

Definition at line 103 of file yonics.hpp.

3.13.2 Constructor & Destructor Documentation

3.13.2.1 SaveSD()

`SaveSD::SaveSD ( )`

Definition at line 3 of file SaveSD.cpp.

3.13.3 Member Function Documentation

3.13.3.1 addFlashOp()

```
bool SaveSD::addFlashOp (
            FlashOp * flash )
```

Definition at line 9 of file SaveSD.cpp.

3.13.3.2 savenow()

```
bool SaveSD::savenow ( )
```

Definition at line 17 of file SaveSD.cpp.

Here is the call graph for this function:

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

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

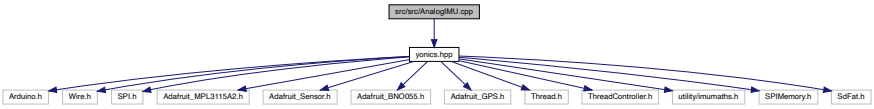
Chapter 4

File Documentation

4.1 src/src/AnalogIMU.cpp File Reference

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

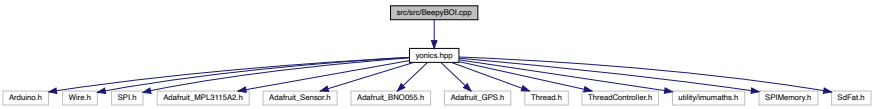
Include dependency graph for AnalogIMU.cpp:



4.2 src/src/BeepyBOI.cpp File Reference

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

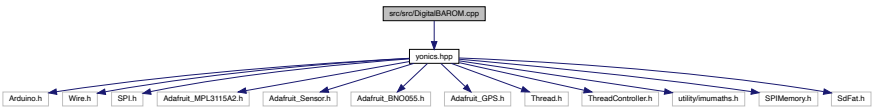
Include dependency graph for BeepyBOI.cpp:



4.3 src/src/DigitalBAROM.cpp File Reference

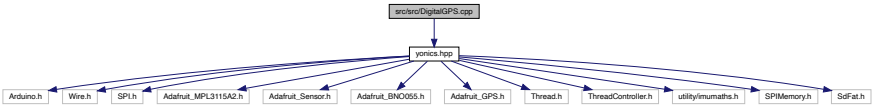
```
#include "yonics.hpp"
```

Include dependency graph for DigitalBAROM.cpp:



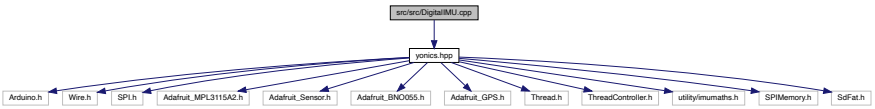
4.4 src/src/DigitalGPS.cpp File Reference

```
#include <yonics.hpp>
```
Include dependency graph for DigitalGPS.cpp:



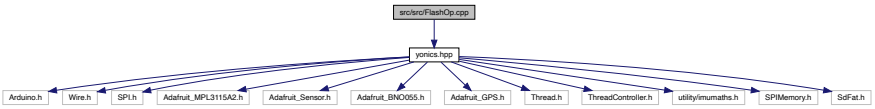
4.5 src/src/DigitalIMU.cpp File Reference

```
#include "yonics.hpp"
```
Include dependency graph for DigitalIMU.cpp:



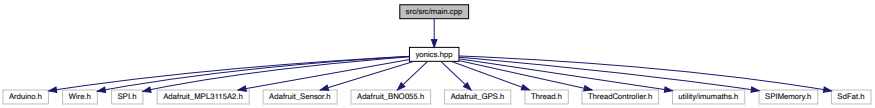
4.6 src/src/FlashOp.cpp File Reference

```
#include "yonics.hpp"
```
Include dependency graph for FlashOp.cpp:



4.7 src/src/main.cpp File Reference

```
#include "yonics.hpp"
```
Include dependency graph for main.cpp:



Macros

- #define GPSSerial Serial3
- #define GPSECHO false

Functions

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

Variables

- int flashWP = 10
- int flashPin = 29
- int speakerPin = 36
- int highG_xPin = 33
- int highG_yPin = 34
- int highG_zPin = 35
- int interval_GPS = 1/10
- int interval_IMU = 40
- int interval_BAROM = 1500
- int interval_ACCEL = 40
- ThreadController thread_control = ThreadController()
- Thread * ThreadGPS = new Thread()
- Thread * ThreadIMU = new Thread()
- Thread * ThreadBAROM = new Thread()
- Thread * ThreadACCEL = new Thread()
- DigitalIMU IMU = DigitalIMU(55,0x28)
- DigitalBAROM BAROM
- AnalogIMU HIGHG = AnalogIMU(highG_xPin,highG_yPin,highG_zPin,true)
- GPSdata gps_data
- IMUdata imu_data
- BAROMdata barom_data
- Acceldata accel_data
- BeepyBOI berp = BeepyBOI(speakerPin)
- uint32_t GPSDataSize
- uint32_t imuDataSize
- uint32_t baromDataSize
- uint32_t accelDataSize
- SaveSD saver
- DigitalGPS * gps_ptr

4.7.1 Macro Definition Documentation

4.7.1.1 GPSECHO

```
#define GPSECHO false
```

Definition at line 47 of file main.cpp.

4.7.1.2 GPSSerial

```
#define GPSSerial Serial3
```

Definition at line 46 of file main.cpp.

4.7.2 Function Documentation

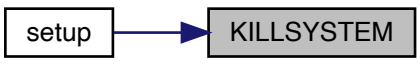
4.7.2.1 KILLSYSTEM()

```
void KILLSYSTEM ( )
```

Definition at line 112 of file main.cpp.

Here is the call graph for this function:

Here is the caller graph for this function:



4.7.2.2 loop()

```
void loop ( )
```

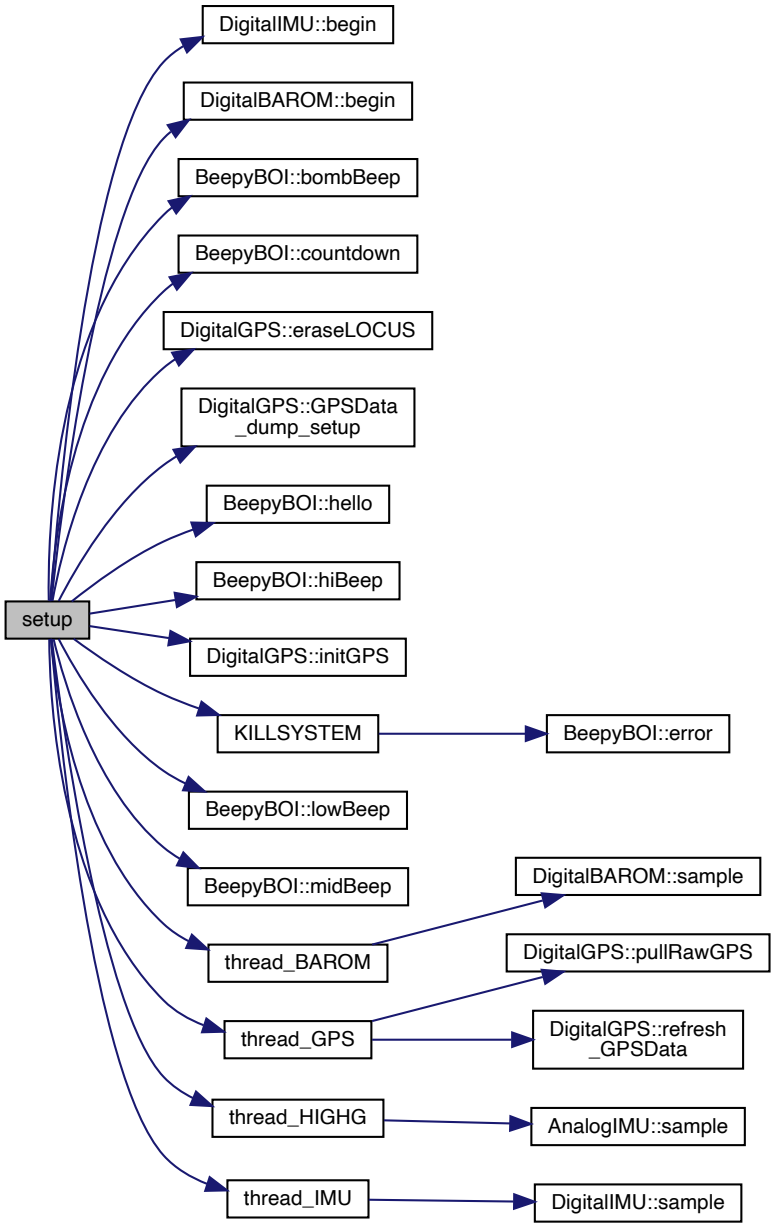
Definition at line 212 of file main.cpp.

4.7.2.3 setup()

```
void setup ( )
```

Definition at line 119 of file main.cpp.

Here is the call graph for this function:

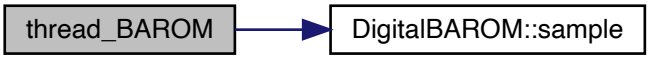


4.7.2.4 thread_BAROM()

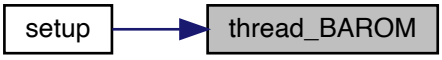
```
void thread_BAROM ( )
```

Definition at line 95 of file main.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

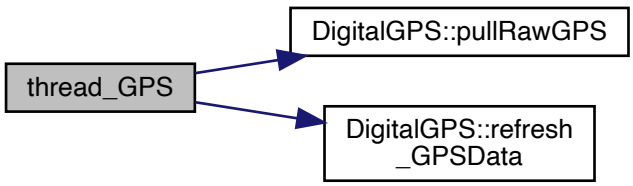


4.7.2.5 thread_GPS()

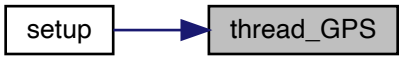
```
void thread_GPS ( )
```

Definition at line 78 of file main.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

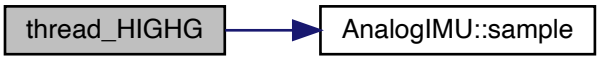


4.7.2.6 thread_HIGHG()

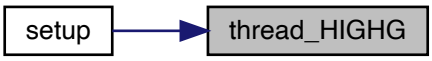
```
void thread_HIGHG ( )
```

Definition at line 103 of file main.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

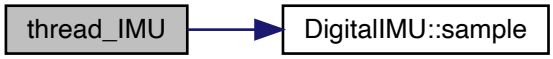


4.7.2.7 thread_IMU()

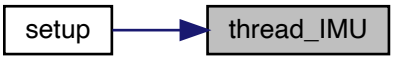
```
void thread_IMU ( )
```

Definition at line 85 of file main.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.7.3 Variable Documentation

4.7.3.1 accel_data

```
Acceldata accel_data
```

Definition at line 59 of file main.cpp.

4.7.3.2 accelDataSize

```
uint32_t accelDataSize
```

Definition at line 73 of file main.cpp.

4.7.3.3 BAROM

```
DigitalBAROM BAROM
```

Definition at line 51 of file main.cpp.

4.7.3.4 barom_data

```
BAROMdata barom_data
```

Definition at line 58 of file main.cpp.

4.7.3.5 baromDataSize

```
uint32_t baromDataSize
```

Definition at line 72 of file main.cpp.

4.7.3.6 berp

```
BeepyBOI berp = BeepyBOI(speakerPin)
```

Definition at line 62 of file main.cpp.

4.7.3.7 flashPin

```
int flashPin = 29
```

Definition at line 20 of file main.cpp.

4.7.3.8 flashWP

```
int flashWP = 10
```

Definition at line 19 of file main.cpp.

4.7.3.9 gps_data

```
GPSdata gps_data
```

Definition at line 56 of file main.cpp.

4.7.3.10 gps_ptr

```
DigitalGPS* gps_ptr
```

Definition at line 76 of file main.cpp.

4.7.3.11 GPSDataSize

```
uint32_t GPSDataSize
```

Definition at line 70 of file main.cpp.

4.7.3.12 HIGHG

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

Definition at line 52 of file main.cpp.

4.7.3.13 highG_xPin

```
int highG_xPin = 33
```

Definition at line 22 of file main.cpp.

4.7.3.14 highG_yPin

```
int highG_yPin = 34
```

Definition at line 23 of file main.cpp.

4.7.3.15 highG_zPin

```
int highG_zPin = 35
```

Definition at line 24 of file main.cpp.

4.7.3.16 IMU

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

Definition at line 50 of file main.cpp.

4.7.3.17 imu_data

```
IMUdata imu_data
```

Definition at line 57 of file main.cpp.

4.7.3.18 imuDataSize

```
uint32_t imuDataSize
```

Definition at line 71 of file main.cpp.

4.7.3.19 interval_ACCEL

```
int interval_ACCEL = 40
```

Definition at line 30 of file main.cpp.

4.7.3.20 interval_BAROM

```
int interval_BAROM = 1500
```

Definition at line 29 of file main.cpp.

4.7.3.21 interval_GPS

```
int interval_GPS = 1/10
```

Definition at line 27 of file main.cpp.

4.7.3.22 interval_IMU

```
int interval_IMU = 40
```

Definition at line 28 of file main.cpp.

4.7.3.23 saver

```
SaveSD saver
```

Definition at line 75 of file main.cpp.

4.7.3.24 speakerPin

```
int speakerPin = 36
```

Definition at line 21 of file main.cpp.

4.7.3.25 thread_control

```
ThreadController thread_control = ThreadController()
```

Definition at line 35 of file main.cpp.

4.7.3.26 ThreadACCEL

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

Definition at line 41 of file main.cpp.

4.7.3.27 ThreadBAROM

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

Definition at line 40 of file main.cpp.

4.7.3.28 ThreadGPS

```
Thread* ThreadGPS = new Thread()
```

Definition at line 38 of file main.cpp.

4.7.3.29 ThreadIMU

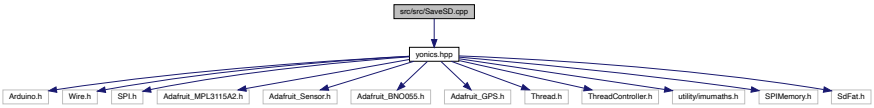
```
Thread* ThreadIMU = new Thread()
```

Definition at line 39 of file main.cpp.

4.8 src/src/SaveSD.cpp File Reference

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

Include dependency graph for SaveSD.cpp:



4.9 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 <SPIMemory.h>
```

```
#include <SdFat.h>
```

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
- struct ourTypes
- struct event
- class FlashOp
- class SaveSD
- class AnalogIMU
- class DigitalIMU
- class DigitalBAROM
- class DigitalGPS
- class BeepyBOI