

# RFzero Webinar

Intermediate

2020-06-20, 20 UTC

# Agenda

- Understanding the structure and flow of the example programs
  - Generic RFzero programs
    - The flow and functional blocks
    - The RFzero H/W interaction
  - Template
  - GPSDO
- The Arduino yield() function
- The RFzero libraries
- Understanding the display functions in display.cpp/~.h

# The format of the presentation

- The Arduino overview
- Down into the generic details
  - The RFzero H/W and S/W overview
  - The RFzero program structure
  - Further down into the program flow
  - Into the details of the functional blocks
  - The EEPROM, Si5351A and GPS library
- Back up again and actual programs
- Down again into yield() and libraries
- Back into the GPSSDO display.cpp

Taking apart

Putting together

Taking apart

Taking apart

# Arduino program structure

MyArduinoProgram.ino

```
#include "extraFunctions.h"  
objects  
constants  
variables
```

```
void setup()  
{  
  runs at start and only one time  
}
```

```
void loop()  
{  
  runs after setup() and runs  
  continuously  
}
```

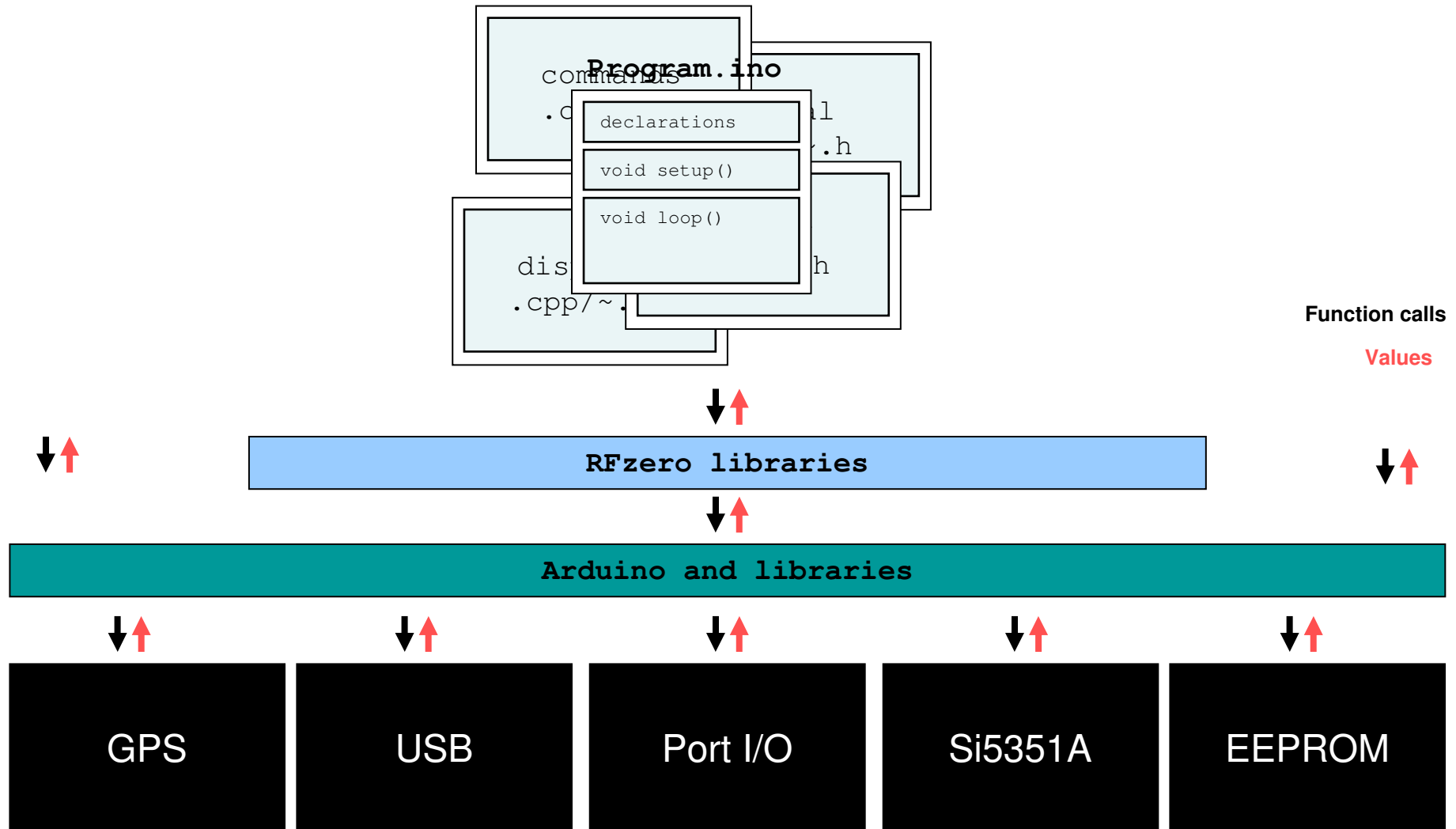
extraFunctions.h

```
#includes ...  
  
Names of functions in  
extraFunctions.cpp  
void functionA(int x);
```

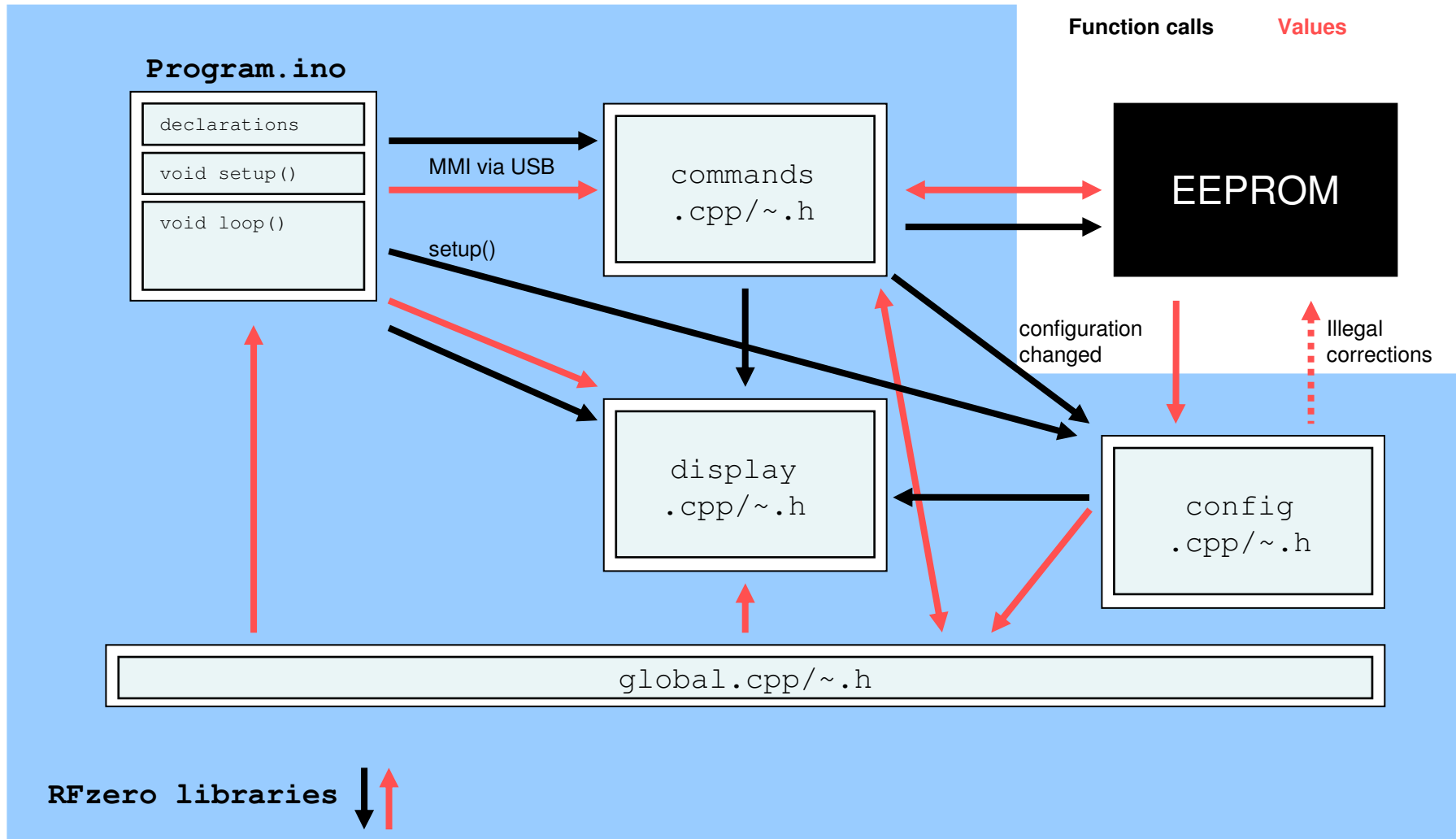
extraFunctions.cpp

```
#include "extraFunctions.h"  
  
objects  
  
constants  
variable  
  
void functionA(int x)  
{  
  
}
```

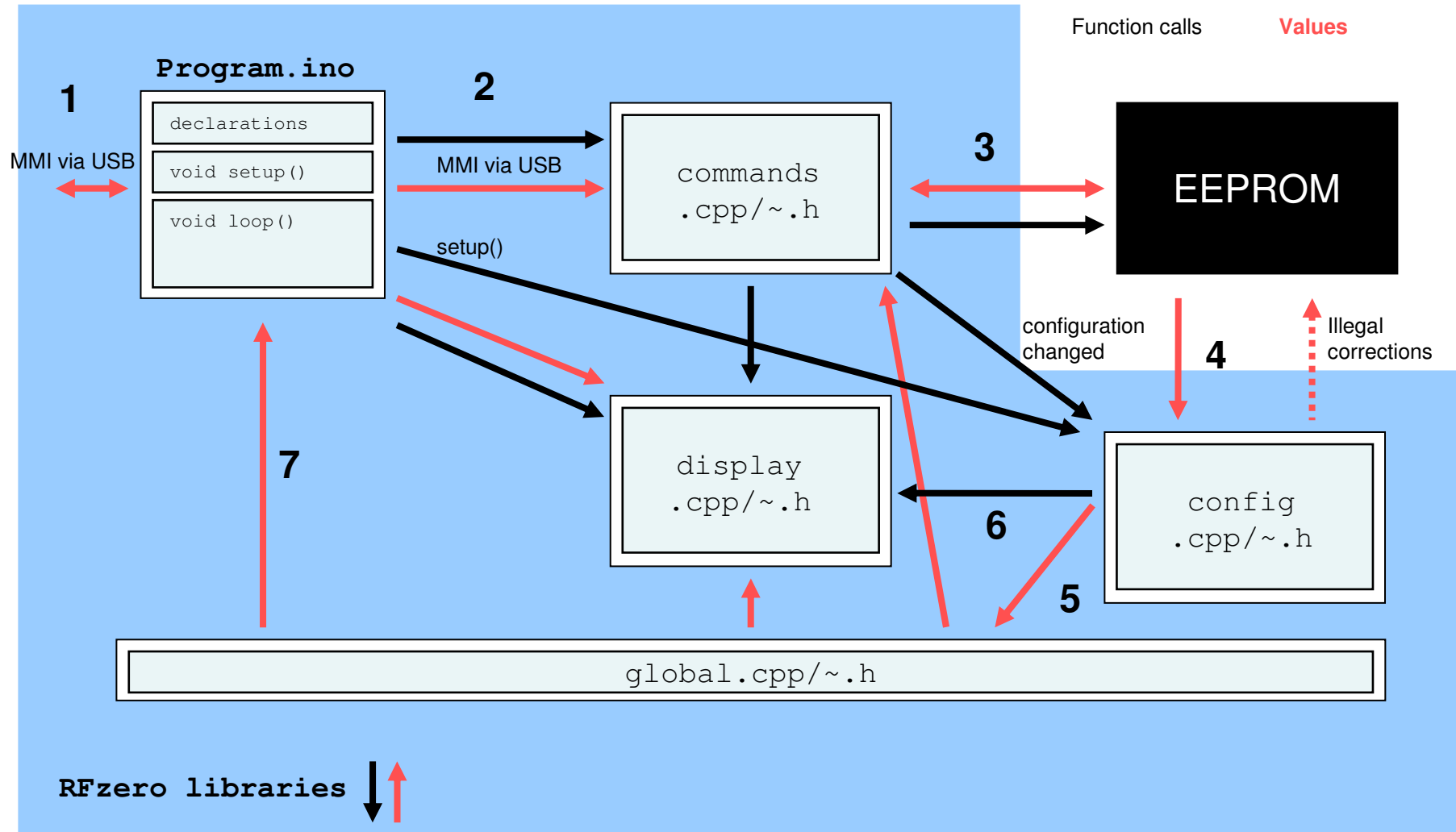
# RFzero H/W + S/W structure



# RFzero program structure



# MMI configuration flow



# Inside the commands.cpp

```
configMode = 0
configChanged = 1
str = trim(string)

If configMode = 0 // run mode MMIs
    if str = "config"
        configMode = 1
else // Below is config mode
    if str = "exit"
        load parameters from EEPROM
    else if str = "freq"
        write freq value to EEPROM
        configChanged = 1
    else if str = "display"
        write display value to EEPROM
        configChanged = 1
...
```

- Receives MMI strings
- Trims received strings
- Checks if in run or config. mode
- Checks MMI and acts if valid
- When "exit" is received and config. was changed calls function in config.cpp to load parameters



# Inside the config.cpp

```
display = eepromRead(display
address)
is display value invalid?
    writeEEPROM(valid display value)

frequency = eepromRead(display
address)
...
setFrequency(frequency)

Display_UpdateFrequency
...
```

- Loads parameters from the EEPROM
- Relies on the EEPROM map
- Checks to see if parameter is valid, and if not writes a valid value to the EEPROM
- Sets parameters and updates the display

# Inside the display.cpp

```
Initialize display

Display_GPSUpdate
  get GPS data

  if displayMode = 1
    ...
  else if displayMode = 2
    cursorXY 10, 0
    print utcSeconds
    cursorXY 7, 0
    print utcMinutes
    cursorXY 4, 0
    print utcHours

  ...
```

- Initializes the display
- Update function(s) must be called whenever there is a change
- Update functions handles display differences
- Parameters are often global or GPS variables, otherwise they are passed as parameters

# EEPROM, Si5351A and GPS

- The EEPROM is only used to store and retrieve configuration parameters
- The Si5351A is setup and managed continuously (math calibrated)
- The GPS manages itself while providing data and gate signal to the frequency counter in the MCU

# Program .ino controls the flow

- In the RFzero programs the .ino file controls the flow by design but not an Arduino requirement
- The Template program
  - Simple overview
- The GPSDO program
  - Simple flow
- The signal generator program
  - Heavy user interaction
- The beacon programs
  - Strict flow

# The Arduino `yield()` function

- Little known by Arduino people
- Runs always also inside `delay()` loops
- Functionality can be added by the programmer
- Can be used for serial, I2C and SPI and USB reception
  - USB must be visited often, but easy with `yield()`
- Must be called in slow loops. Otherwise the program will see the program as unresponsive

# The RFzero libraries

- The RFzero main library (RFzero.h)
  - EEPROM: eeprom.~
  - Frequency counter: freqCount.~
  - GPS: gps.~
  - MCU: mcu.~
  - RFzero: RFzero.~
  - Si5351A: si5351a.~
- The RFzero Modes library
  - Modes: modes.~
- The RFzero utilities library
- The RFzero LCD I2C library

# Functions in display.cpp

- void Display\_StartScreen(bool splash)
- void Display\_GPSUpdate()
- ...
- Don't update unless there is a need to.  
Displays are slow devices