DroneIno

Generated by Doxygen 1.9.3

Chapter 1

Todo List

Member myISR ()

Use pwm to read the receivers signals, avoid digitalRead.

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

Class Documentation

4.1 jpg_chunking_t Struct Reference

Public Attributes

- httpd_req_t * req
- size_t len

4.1.1 Member Data Documentation

4.1.1.1 len

```
size_t jpg_chunking_t::len
```

4.1.1.2 req

```
httpd_req_t* jpg_chunking_t::req
```

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

- WiFiTelemetry/app_httpd.cpp

4.2 ra_filter_t Struct Reference

Public Attributes

- size_t size
- size_t index
- size_t count
- int sum
- int * values

4.2.1 Member Data Documentation

4.2.1.1 count

```
size_t ra_filter_t::count
```

4.2.1.2 index

```
size_t ra_filter_t::index
```

4.2.1.3 size

```
size_t ra_filter_t::size
```

4.2.1.4 sum

```
int ra_filter_t::sum
```

4.2.1.5 values

```
int* ra_filter_t::values
```

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

- WiFiTelemetry/app_httpd.cpp

4.3 trimPosition Struct Reference

NOT DECLARED VARIABLES.

```
#include <Globals.h>
```

Public Attributes

- byte reverse
- int16_t low
- int16_t center
- int16_t high
- int16_t actual

4.3.1 Detailed Description

NOT DECLARED VARIABLES.

4.3.2 Member Data Documentation

4.3.2.1 actual

```
int16_t trimPosition::actual
```

4.3.2.2 center

```
int16_t trimPosition::center
```

4.3.2.3 high

```
int16_t trimPosition::high
```

4.3.2.4 low

```
int16_t trimPosition::low
```

4.3.2.5 reverse

```
byte trimPosition::reverse
```

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

- DroneIno/Globals.h

Chapter 5

File Documentation

5.1 Calibration/Calibration.ino File Reference

Calibration sketch for the motors, sensors and receiver inputs.

```
#include <Arduino.h>
#include <Wire.h>
#include <EEPROM.h>
```

Macros

- #define BAUD_RATE 115200
- #define EEPROM_SIZE 36
- #define WIRE_CLOCK 400000L
- #define PIN_ESC_1 17
- #define PIN_ESC_2 16
- #define PIN_ESC_3 27
- #define PIN_ESC_4 14
- #define PIN_RECEIVER_1 12
- #define PIN_RECEIVER_2 13
- #define PIN_RECEIVER_3 5
- #define PIN_RECEIVER_4 23
- #define PIN_BATTERY_LED 19
- #define PIN_DIGITAL_13 18
- #define PIN_BATTERY_LEVEL 4
- #define PIN_PROXIMITY_SENSOR_ECHO 26
- #define PIN_PROXIMITY_SENSOR_TRIG 25
- #define PIN_ALTITUDE_SENSOR_SDA 36
- #define PIN_ALTITUDE_SENSOR_SCL 39

Functions

- void setup ()
- void loop ()
- void myISR ()

Variables

- byte lastChannel1
- byte lastChannel2
- byte lastChannel3
- byte lastChannel4
- byte eepromData [36]
- byte start
- byte data
- boolean new_function_request
- boolean firstAngle
- volatile int receiverInputChannel1
- volatile int receiverInputChannel2
- volatile int receiverInputChannel3
- volatile int receiverInputChannel4
- int esc_1
- int esc_2
- int esc_3
- int esc_4
- int counterChannel1
- int counterChannel2
- int counterChannel3
- int counterChannel4
- int receiverInput [5]
- int loopCounter
- int gyroAddress
- int vibrationCounter
- long accX
- long accY
- long accZ
- long accTotalVector [20]
- long accAvVector
- long vibrationTotalResult
- unsigned long timerChannel1
- unsigned long timerChannel2
- unsigned long timerChannel3
- unsigned long timerChannel4
- unsigned long escTimer
- unsigned long escLoopTimer
- unsigned long zeroTimer
- unsigned long timer1
- unsigned long timer2
- unsigned long timer3
- unsigned long timer4
- unsigned long currentTime
- float angleRollAcc
- float anglePitchAcc
- float anglePitch
- float angleRoll
- int calInt
- int16_t accAxis [4]
- int16_t gyroAxis [4]
- int16_t temperature
- int32_t gyroAxisCal [4]
- int32_t accAxisCcal [4]

- float fromATmega32ToEsp32 = 240 / 16
- float accCorrection = 1
- float gyroCorrection = 1
- const int gyroFrequency = 250
- const float gyroSensibility = 65.5
- float travelCoeff = 1/((float)gyroFrequency ∗ gyroSensibility)
- float convDegToRad = 180.0 / PI
- float travelCoeffToRad = travelCoeff / convDegToRad

5.1.1 Detailed Description

Calibration sketch for the motors, sensors and receiver inputs.

Author

@sebastiano123-c

Version

0.1

Date

2022-02-28

TERMS OF USE

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Note

Always remove the propellers and stay away from the motors unless you are 100% certain of what you are doing.

Dial these on the serial:

- r: print receiver signals.
- a: print quadcopter angles.
- 1: check rotation / vibrations for motor 1 (right front CCW).
- 2: check rotation / vibrations for motor 2 (right rear CW).
- 3: check rotation / vibrations for motor 3 (left rear CCW).
- 4: check rotation / vibrations for motor 4 (left front CW).
- 5: check vibrations for all motors together.

Copyright

5.1.2 Macro Definition Documentation

5.1.2.1 BAUD_RATE

```
#define BAUD_RATE 115200
```

5.1.2.2 EEPROM_SIZE

```
#define EEPROM_SIZE 36
```

5.1.2.3 PIN_ALTITUDE_SENSOR_SCL

```
#define PIN_ALTITUDE_SENSOR_SCL 39
```

5.1.2.4 PIN_ALTITUDE_SENSOR_SDA

```
#define PIN_ALTITUDE_SENSOR_SDA 36
```

5.1.2.5 PIN_BATTERY_LED

```
#define PIN_BATTERY_LED 19
```

5.1.2.6 PIN_BATTERY_LEVEL

```
#define PIN_BATTERY_LEVEL 4
```

5.1.2.7 PIN_DIGITAL_13

```
#define PIN_DIGITAL_13 18
```

5.1.2.8 PIN_ESC_1

```
#define PIN_ESC_1 17
```

5.1.2.9 PIN_ESC_2

```
#define PIN_ESC_2 16
```

5.1.2.10 PIN_ESC_3

```
#define PIN_ESC_3 27
```

5.1.2.11 PIN_ESC_4

```
#define PIN_ESC_4 14
```

5.1.2.12 PIN_PROXIMITY_SENSOR_ECHO

```
#define PIN_PROXIMITY_SENSOR_ECHO 26
```

5.1.2.13 PIN_PROXIMITY_SENSOR_TRIG

```
#define PIN_PROXIMITY_SENSOR_TRIG 25
```

5.1.2.14 PIN_RECEIVER_1

```
#define PIN_RECEIVER_1 12
```

5.1.2.15 PIN_RECEIVER_2

```
#define PIN_RECEIVER_2 13
```

5.1.2.16 PIN_RECEIVER_3

```
#define PIN_RECEIVER_3 5
```

5.1.2.17 PIN_RECEIVER_4

```
#define PIN_RECEIVER_4 23
```

5.1.2.18 WIRE_CLOCK

```
#define WIRE_CLOCK 400000L
```

5.1.3 Function Documentation

5.1.3.1 loop()

```
void loop ( )
```

5.1.3.2 myISR()

```
void myISR ( )
```

5.1.3.3 setup()

```
void setup ( )
```

5.1.4 Variable Documentation

5.1.4.1 accAvVector

```
long accAvVector
```

5.1.4.2 accAxis

```
int16_t accAxis[4]
```

5.1.4.3 accAxisCcal

```
int32_t accAxisCcal[4]
```

5.1.4.4 accCorrection

```
float accCorrection = 1
```

5.1.4.5 accTotalVector

```
long accTotalVector[20]
```

5.1.4.6 accX

```
long accX
```

5.1.4.7 accY

```
long accY
```

5.1.4.8 accZ

```
long accZ
```

5.1.4.9 anglePitch

```
float anglePitch
```

5.1.4.10 anglePitchAcc

```
float anglePitchAcc
```

5.1.4.11 angleRoll

```
float angleRoll
```

5.1.4.12 angleRollAcc

```
float angleRollAcc
```

5.1.4.13 calInt

```
int calInt
```

5.1.4.14 convDegToRad

```
float convDegToRad = 180.0 / PI
```

5.1.4.15 counterChannel1

```
int counterChannel1
```

5.1.4.16 counterChannel2

```
int counterChannel2
```

5.1.4.17 counterChannel3

```
int counterChannel3
```

5.1.4.18 counterChannel4

```
int counterChannel4
```

5.1.4.19 currentTime

```
unsigned long currentTime
```

5.1.4.20 data

```
byte data
```

5.1.4.21 eepromData

```
byte eepromData[36]
```

5.1.4.22 esc_1

```
int esc_1
```

5.1.4.23 esc_2

```
int esc_2
```

5.1.4.24 esc_3

```
int esc_3
```

5.1.4.25 esc_4

```
int esc_4
```

5.1.4.26 escLoopTimer

```
unsigned long escLoopTimer
```

5.1.4.27 escTimer

```
unsigned long escTimer
```

5.1.4.28 firstAngle

```
boolean firstAngle
```

5.1.4.29 fromATmega32ToEsp32

```
float fromATmega32ToEsp32 = 240 / 16
```

5.1.4.30 gyroAddress

```
int gyroAddress
```

5.1.4.31 gyroAxis

```
int16_t gyroAxis[4]
```

5.1.4.32 gyroAxisCal

```
int32_t gyroAxisCal[4]
```

5.1.4.33 gyroCorrection

```
float gyroCorrection = 1
```

5.1.4.34 gyroFrequency

```
const int gyroFrequency = 250
```

5.1.4.35 gyroSensibility

```
const float gyroSensibility = 65.5
```

5.1.4.36 lastChannel1

```
byte lastChannel1
```

5.1.4.37 lastChannel2

```
byte lastChannel2
```

5.1.4.38 lastChannel3

```
byte lastChannel3
```

5.1.4.39 lastChannel4

```
byte lastChannel4
```

5.1.4.40 loopCounter

```
int loopCounter
```

5.1.4.41 new_function_request

```
boolean new_function_request
```

5.1.4.42 receiverInput

```
int receiverInput[5]
```

5.1.4.43 receiverInputChannel1

```
volatile int receiverInputChannel1
```

5.1.4.44 receiverInputChannel2

```
volatile int receiverInputChannel2
```

5.1.4.45 receiverInputChannel3

```
volatile int receiverInputChannel3
```

5.1.4.46 receiverInputChannel4

```
volatile int receiverInputChannel4
```

5.1.4.47 start

```
byte start
```

5.1.4.48 temperature

```
int16_t temperature
```

5.1.4.49 timer1

```
unsigned long timer1
```

5.1.4.50 timer2

```
unsigned long timer2
```

5.1.4.51 timer3

```
unsigned long timer3
```

5.1.4.52 timer4

```
unsigned long timer4
```

5.1.4.53 timerChannel1

```
unsigned long timerChannel1
```

5.1.4.54 timerChannel2

```
unsigned long timerChannel2
```

5.1.4.55 timerChannel3

```
unsigned long timerChannel3
```

5.1.4.56 timerChannel4

```
unsigned long timerChannel4
```

5.1.4.57 travelCoeff

```
float travelCoeff = 1/((float)gyroFrequency * gyroSensibility)
```

5.1.4.58 travelCoeffToRad

```
float travelCoeffToRad = travelCoeff / convDegToRad
```

5.1.4.59 vibrationCounter

```
int vibrationCounter
```

5.1.4.60 vibrationTotalResult

```
long vibrationTotalResult
```

5.1.4.61 zeroTimer

```
unsigned long zeroTimer
```

5.2 Calibration/checkGyro.ino File Reference

Gyroscope routines.

Macros

- #define GYRO_CONFIG 0x1B
- #define ACCEL_CONFIG 0x1C
- #define ACCEL_XOUT_H 0x3B
- #define ACCEL_YOUT_H 0x3D
- #define ACCEL_ZOUT_H 0x3F
- #define GYRO_XOUT_H 0x43
- #define GYRO_YOUT_H 0x45
- #define GYRO_ZOUT_H 0x47
- #define PWR_MGMT_1 0x6B
- #define PWR_MGMT_2 0x6C
- #define BOUND_POSITION 4096

Functions

- void ymfcFunction (bool PRY=true, bool XYZ=false)

  Calculate the pitch and roll angle.
- void checkGyro ()

  Main routine.
- void calibrateGyro ()
- void getGyroSignal ()
- void getAcc ()
- void getGyro ()
- void setupMPU ()
- void setGyroRegisters ()

Variables

- byte rawAX [2]
- byte rawAY [2]
- byte rawAZ [2]
- byte rawGX [2]
- byte rawGY [2]
- byte rawGZ [2]

5.2.1 Detailed Description

Gyroscope routines.

Author

@sebastiano123-c

Version

0.1

Date

2022-03-01

Copyright

Copyright (c) 2022

5.2.2 Macro Definition Documentation

5.2.2.1 ACCEL_CONFIG

```
#define ACCEL_CONFIG 0x1C
```

5.2.2.2 ACCEL_XOUT_H

```
#define ACCEL_XOUT_H 0x3B
```

5.2.2.3 ACCEL_YOUT_H

```
#define ACCEL_YOUT_H 0x3D
```

5.2.2.4 ACCEL_ZOUT_H

```
#define ACCEL_ZOUT_H 0x3F
```

5.2.2.5 BOUND_POSITION

```
#define BOUND_POSITION 4096
```

5.2.2.6 GYRO_CONFIG

```
#define GYRO_CONFIG 0x1B
```

5.2.2.7 GYRO_XOUT_H

```
#define GYRO_XOUT_H 0x43
```

5.2.2.8 GYRO_YOUT_H

```
#define GYRO_YOUT_H 0x45
```

5.2.2.9 GYRO_ZOUT_H

```
#define GYRO_ZOUT_H 0x47
```

5.2.2.10 PWR_MGMT_1

```
#define PWR_MGMT_1 0x6B
```

5.2.2.11 PWR_MGMT_2

```
#define PWR_MGMT_2 0x6C
```

5.2.3 Function Documentation

5.2.3.1 calibrateGyro()

```
void calibrateGyro ( )
```

5.2.3.2 checkGyro()

```
void checkGyro ( )
```

Main routine.

We don't want the esc's to be beeping annoyingly. So let's give them a 1000us puls while calibrating the gyro.

5.2.3.3 getAcc()

```
void getAcc ( )
```

5.2.3.4 getGyro()

```
void getGyro ( )
```

5.2.3.5 getGyroSignal()

```
void getGyroSignal ( )
```

5.2.3.6 setGyroRegisters()

```
void setGyroRegisters ( )
```

5.2.3.7 setupMPU()

```
void setupMPU ( )
```

5.2.3.8 ymfcFunction()

```
void ymfcFunction (
            bool PRY = true,
            bool XYZ = false )
```

Calculate the pitch and roll angle.

Parameters

PRY	
XYZ	

5.2.4 Variable Documentation

5.2.4.1 rawAX

```
byte rawAX[2]
```

5.2.4.2 rawAY

```
byte rawAY[2]
```

5.2.4.3 rawAZ

```
byte rawAZ[2]
```

5.2.4.4 rawGX

```
byte rawGX[2]
```

5.2.4.5 rawGY

```
byte rawGY[2]
```

5.2.4.6 rawGZ

```
byte rawGZ[2]
```

5.3 Calibration/receiverRoutines.ino File Reference

Receiver routines.

Functions

- void waitForReceiver ()

  Check if the receiver values are valid within 10 seconds.
- int convertReceiverChannel (byte function)

  Convert the receiver signal.
- void printSignals ()

  Prints the signals.
- void rFunction ()

  Main routine used in the sketch.

5.3.1 Detailed Description

Receiver routines.

Author

@sebastiano123-c

Version

0.1

Date

2022-03-01

Copyright

5.3.2 Function Documentation

5.3.2.1 convertReceiverChannel()

```
int convertReceiverChannel (
            byte function )
```

Convert the receiver signal.

This part converts the actual receiver signals to a standardized 1000 2000us value. The stored data in the EEPROM is used.

Parameters

function	

Returns

int

5.3.2.2 printSignals()

```
void printSignals ( )
```

Prints the signals.

5.3.2.3 rFunction()

```
void rFunction ( )
```

Main routine used in the sketch.

5.3.2.4 waitForReceiver()

```
void waitForReceiver ( )
```

Check if the receiver values are valid within 10 seconds.

5.4 DroneIno/Altitude.ino File Reference

Pressure readings routines.

Functions

- void checkAltitudeSensor ()
  Check if the altitude sensor is connected with I2C.
- void readTrim ()
  Read barometer data from I2C communication.
- void writeRegister (uint8_t reg_address, uint8_t msg)
  Writes a message to the I2C designed address.
- void readPressureData ()
  Reads pressure data.
- signed long int calibration_T (signed long int adc_T)
  Calibrates temperature readings.
- unsigned long int calibration_P (signed long int adc_P)
  Calibrates pressure readings.
- void smoothPressureReadings ()
  Filter barometer readings for a smoother flight.
- void calculateAltitudeAdjustementPID ()
  Transforms the barometer readings into the PID output pulses.
- void calculateAltitudeHold ()
  Main routine governing the barometer acquisitions.

5.4.1 Detailed Description

Pressure readings routines.

Author

@sebastiano123-c

Depending on the ALTITUDE_SENSOR macro value:

- BMP280: uses the I2C communication WITHOUT using the ADAFRUIT library to better performances;
- BME280: not yet implemented;
- OFF: no altitude sensor, so there is no pressure data acquisition.

Note

For now, the only sensor available is the BMP280, otherwise these routines return void.

Version

0.1

Date

2022-02-18

Copyright

5.4.2 Function Documentation

5.4.2.1 calculateAltitudeAdjustementPID()

```
void calculateAltitudeAdjustementPID ( )
```

Transforms the barometer readings into the PID output pulses.

When the throttle stick position is increased or decreased the altitude hold function is partially disabled. The manualAltitudeChange variable will indicate if the altitude of the quadcopter is changed by the pilot.

5.4.2.2 calculateAltitudeHold()

```
void calculateAltitudeHold ( )
```

Main routine governing the barometer acquisitions.

5.4.2.3 calibration_P()

```
unsigned long int calibration_P (
            signed long int adc_P )
```

Calibrates pressure readings.

Parameters

adc↩ _P	

Returns

unsigned long int

5.4.2.4 calibration_T()

```
signed long int calibration_T (
             signed long int adc_T )
```

Calibrates temperature readings.

Parameters

adc↩ _T	

Returns

signed long int

5.4.2.5 checkAltitudeSensor()

```
void checkAltitudeSensor ( )
```

Check if the altitude sensor is connected with I2C.

5.4.2.6 readPressureData()

```
void readPressureData ( )
```

Reads pressure data.

5.4.2.7 readTrim()

```
void readTrim ( )
```

Read barometer data from I2C communication.

5.4.2.8 smoothPressureReadings()

```
void smoothPressureReadings ( )
```

Filter barometer readings for a smoother flight.

5.4.2.9 writeRegister()

```
void writeRegister (
            uint8_t reg_address,
            uint8_t msg )
```

Writes a message to the I2C designed address.

Parameters

reg_address	
msg	

5.5 DroneIno/Battery.ino File Reference

Battery calculations routines.

Functions

- float voltagePartitor (int volt)

  Calculates the voltage partitor given the total drop defined in the Constants.ino.
- float fromWidthToVPin (int width, float corr=1.)

  Calculates the the coversion between the pin readings and the voltage.
- float fromWidthToVBattery (int width, float corr=1.)

  Calculates the conversion between the analog pin readings and the battery initial voltage.
- void initBattery ()

  Get the initial value of the battery voltage.
- int getBatteryVoltage ()

  Get battery voltage.
- void batteryVoltageCompensation ()

  Compensate the ESCs pulses with battery voltage.

5.5.1 Detailed Description

Battery calculations routines.

Author

@sebastiano123-c

Version

0.1

Date

2022-02-28

Copyright

5.5.2 Function Documentation

5.5.2.1 batteryVoltageCompensation()

```
void batteryVoltageCompensation ( )
```

Compensate the ESCs pulses with battery voltage.

correctionBattery/maxBatteryLevelDropped;

5.5.2.2 fromWidthToVBattery()

```
float fromWidthToVBattery (
            int width,
            float corr = 1.  )
```

Calculates the conversion between the analog pin readings and the battery initial voltage.

Parameters

width	
corr	

Returns

float

5.5.2.3 fromWidthToVPin()

```
float fromWidthToVPin (
            int width,
            float corr = 1.  )
```

Calculates the the coversion between the pin readings and the voltage.

Parameters

width	
corr	

Returns

float

5.5.2.4 getBatteryVoltage()

```
int getBatteryVoltage ( )
```

Get battery voltage.

5.5.2.5 initBattery()

```
void initBattery ( )
```

Get the initial value of the battery voltage.

5.5.2.6 voltagePartitor()

```
float voltagePartitor (
            int volt )
```

Calculates the voltage partitor given the total drop defined in the Constants.ino.

Parameters

volt	

Returns

float

5.6 DroneIno/Config.h File Reference

Configuration file.

Macros

- #define PINMAP ESP32
- #define DEBUG false
- #define BAUD_RATE 115200
- #define WIRE_CLOCK 400000L
- #define EEPROM_SIZE 36
- #define AUTO_LEVELING true
- #define MAX_BATTERY_VOLTAGE 11100
- #define MIN_BATTERY_VOLTAGE 3000
- #define WIFI_TELEMETRY ESP_CAM
- #define GYROSCOPE MPU6050
- #define ALTITUDE_SENSOR BMP280
- #define PROXIMITYSENSOR OFF
- #define DANGER_DISTANCE 40

5.6.1 Detailed Description

Configuration file.

Author

@sebastiano123-c

Note

∗ = not available at the moment

Version

0.1

Date

2022-02-28

Copyright

5.6.2 Macro Definition Documentation

5.6.2.1 ALTITUDE_SENSOR

```
#define ALTITUDE_SENSOR BMP280
```

5.6.2.2 AUTO_LEVELING

```
#define AUTO_LEVELING true
```

5.6.2.3 BAUD_RATE

```
#define BAUD_RATE 115200
```

5.6.2.4 DANGER_DISTANCE

```
#define DANGER_DISTANCE 40
```

5.6.2.5 DEBUG

```
#define DEBUG false
```

5.6.2.6 EEPROM_SIZE

```
#define EEPROM_SIZE 36
```

5.6.2.7 GYROSCOPE

```
#define GYROSCOPE MPU6050
```

5.6.2.8 MAX_BATTERY_VOLTAGE

```
#define MAX_BATTERY_VOLTAGE 11100
```

5.6.2.9 MIN_BATTERY_VOLTAGE

```
#define MIN_BATTERY_VOLTAGE 3000
```

5.6.2.10 PINMAP

```
#define PINMAP ESP32
```

5.6.2.11 PROXIMITYSENSOR

```
#define PROXIMITYSENSOR OFF
```

5.6.2.12 WIFI_TELEMETRY

```
#define WIFI_TELEMETRY ESP_CAM
```

5.6.2.13 WIRE_CLOCK

```
#define WIRE_CLOCK 400000L
```

5.7 Config.h

Go to the documentation of this file.

```
1
14 // MICROCONTROLLER BOARD
15 #define PINMAP                      ESP32                   // (OFF*, ATMEGA32*, ESP32) OFF if the board
       is Arduino Uno (directly write pin)
16
17
18 // DEBUG MODE
19 #define DEBUG                       false                   // (true, false) true enable serial prints
       for debugging
20
21
22 // SKETCH CONSTANTS
23 #define BAUD_RATE                   115200                  // (9600, 57600, 115200)
24 #define WIRE_CLOCK                  400000L                 // (100000L, 400000L) 400000L is suggested
25 #define EEPROM_SIZE                 36                      // eeprom allocatable memory
26
27
28 // AUTO LEVElING
29 #define AUTO_LEVELING               true                    // (true, false) false disable every
       autoleveling compensation
30
31
32 // BATTERY
33 #define MAX_BATTERY_VOLTAGE         11100                   // (mV) battery nominal maximum voltage (use
       ONLY 11.1V batteries)
34 #define MIN_BATTERY_VOLTAGE         3000                    // (mV) battery level above which it is
       dangerous to use
35
36
37 // WIFI
38 #define WIFI_TELEMETRY              ESP_CAM                 // (NATIVE, ESP_CAM*)
39
40
41 // SENSORS
42 #define GYROSCOPE                   MPU6050                 // (MPU6050) unique for now
43 #define ALTITUDE_SENSOR             BMP280                  // (OFF, BMP280, BME280*)
44 #define PROXIMITYSENSOR             OFF                     // (OFF, HCSR04*)
45 #define DANGER_DISTANCE             40                      // (cm) Proximity sensor distance above
       which is considered a danger zone
```

5.8 DroneIno/Constants.h File Reference

DroneIno constants for macros.

Macros

- #define OFF 0
- #define ON 1
- #define ESP32 2
- #define ATMEGA32 3
- #define MPU6050 10
- #define BMP280 11
- #define BME280 12
- #define HCSR04 13
- #define NATIVE 15
- #define ESP_CAM 16

5.8.1 Detailed Description

DroneIno constants for macros.

Author

@sebastiano123-c

Version

0.1

Date

2022-02-28

Copyright

5.8.2 Macro Definition Documentation

5.8.2.1 ATMEGA32

```
#define ATMEGA32 3
```

5.8.2.2 BME280

```
#define BME280 12
```

5.8.2.3 BMP280

```
#define BMP280 11
```

5.8.2.4 ESP32

```
#define ESP32 2
```

5.8.2.5 ESP_CAM

```
#define ESP_CAM 16
```

5.8.2.6 HCSR04

```
#define HCSR04 13
```

5.8.2.7 MPU6050

```
#define MPU6050 10
```

5.8.2.8 NATIVE

```
#define NATIVE 15
```

5.8.2.9 OFF

```
#define OFF 0
```

5.8.2.10 ON

```
#define ON 1
```

5.9 Constants.h

Go to the documentation of this file.

```
1
12 #define OFF                         0
13 #define ON                          1
14
15 #define ESP32                       2
16 #define ATMEGA32                    3
17
18 #define MPU6050                     10
19 #define BMP280                      11
20 #define BME280                      12
21 #define HCSR04                      13
22
23 #define NATIVE                      15
24 #define ESP_CAM                     16
```

5.10 DroneIno/Controller.ino File Reference

Radiocommander control routines.

Functions

- void waitController ()

  Wait until the receiver is active and the throttle is set to the lower position.
- int convertReceiverChannel (byte ch)

  Converts the receiver input signal.

5.10.1 Detailed Description

Radiocommander control routines.

Author

@sebastiano123-c

The radio controller, for example the FlySky, works as PWM signals with variable time widths in the range 1000-2000us. The convertReceiverChannel() routine converts the input signal into the range 1000-2000us. waitController() routine is called in the setup() function and waits until DroneIno recognizes a readable radio controller input.

Version

0.1

Date

2022-02-28

Copyright

5.10.2 Function Documentation

5.10.2.1 convertReceiverChannel()

```
int convertReceiverChannel (
            byte ch )
```

Converts the receiver input signal.

Parameters

ch	

Returns

int

5.10.2.2 waitController()

```
void waitController ( )
```

Wait until the receiver is active and the throttle is set to the lower position.

5.11 DroneIno/DroneIno.ino File Reference

Pilot your ESP32 based DIY drone.

```
#include <Arduino.h>
#include <Wire.h>
#include <EEPROM.h>
#include "Constants.h"
#include <Config.h>
#include "src/Models.h"
#include "Globals.h"
```

Functions

- void setup ()
- void loop ()

5.11.1 Detailed Description

Pilot your ESP32 based DIY drone.

Author

@sebastiano123-c

Version

0.2

Date

2022-02-18

TERMS OF USE

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Note

Always remove the propellers and stay away from the motors unless you are 100% certain of what you are doing.

Copyright

5.11.2 Function Documentation

5.11.2.1 loop()

```
void loop ( )
```

5.11.2.2 setup()

```
void setup ( )
```

5.12 Calibration/ESC.ino File Reference

ESC pulses output routines.

Functions

- void escFunction ()

  Calculate the ESC pulse for the outputs.
- void escPulseOutput ()

  ESC outputs.

5.12.1 Detailed Description

ESC pulses output routines.

Author

@sebastiano123-c

Version

0.1

Date

2022-03-01

Copyright

5.12.2 Function Documentation

5.12.2.1 escFunction()

```
void escFunction ( )
```

Calculate the ESC pulse for the outputs.

5.12.2.2 escPulseOutput()

```
void escPulseOutput ( )
```

ESC outputs.

5.13 DroneIno/ESC.ino File Reference

ESC routines to govern the motors.

Functions

- void droneStart ()

  First settings of the PID and angles.
- void setEscPulses ()

  Set ESC pulses.
- void convertAllSignals ()

  Converts PWM signals input into to the standard 1000 - 2000us.

5.13.1 Detailed Description

ESC routines to govern the motors.

Author

@sebastiano123-c

Version

0.1

Date

2022-02-28

Copyright

5.13.2 Function Documentation

5.13.2.1 convertAllSignals()

```
void convertAllSignals ( )
```

Converts PWM signals input into to the standard 1000 - 2000us.

5.13.2.2 droneStart()

```
void droneStart ( )
```

First settings of the PID and angles.

5.13.2.3 setEscPulses()

```
void setEscPulses ( )
```

Set ESC pulses.

2000∗MAX_DUTY_CYCLE

2000∗MAX_DUTY_CYCLE

2000∗MAX_DUTY_CYCLE

2000∗MAX_DUTY_CYCLE

5.14 DroneIno/Globals.h File Reference

Global variables used in all the project.

Classes

- struct trimPosition

  NOT DECLARED VARIABLES.

Variables

- float PID_P_GAIN_ROLL = 1.0

  PID parameters for roll, pitch, yaw and altitude.
- float PID_I_GAIN_ROLL = 0.000001
- float PID_D_GAIN_ROLL = 10.0
- int PID_MAX_ROLL = 400
- float PID_P_GAIN_PITCH = PID_P_GAIN_ROLL
- float PID_I_GAIN_PITCH = PID_I_GAIN_ROLL
- float PID_D_GAIN_PITCH = PID_D_GAIN_ROLL
- int PID_MAX_PITCH = PID_MAX_ROLL
- float PID_P_GAIN_YAW = 0.7
- float PID_I_GAIN_YAW = 0.05
- float PID_D_GAIN_YAW = 0.0
- int PID_MAX_YAW = 400
- float PID_P_GAIN_ALTITUDE = 1.4
- float PID_I_GAIN_ALTITUDE = 0.3
- float PID_D_GAIN_ALTITUDE = 0.75
- int PID_MAX_ALTITUDE = 400
- float GYROSCOPE_ROLL_FILTER = .9996

  GYROSCOPE.
- float GYROSCOPE_PITCH_FILTER = GYROSCOPE_PITCH_FILTER
- float GYROSCOPE_ROLL_CORR = -.30
- float GYROSCOPE_PITCH_CORR = -1.65
- const int gyroFrequency = 250
- const float gyroSensibility = 65.5
- const int correctionPitchRoll = 15

- float convDegToRad = 180.0 / PI
- float travelCoeff = 1/((float)gyroFrequency * gyroSensibility)
- float travelCoeffToRad = travelCoeff / convDegToRad
- float anglePitchOffset = 0
- float angleRollOffset = 0
- const int pwmLedChannel = 0

  PWM.
- const int pwmChannel1 = 1
- const int pwmChannel2 = 2
- const int pwmChannel3 = 3
- const int pwmChannel4 = 4
- const int pwmLedFlyChannel = 5
- const int freq = 500
- const int resolution = 11
- const int MAX_DUTY_CYCLE = (int)(pow(2, resolution) - 1)
- const int HALF_DUTY_CYCLE = (int)(0.5*MAX_DUTY_CYCLE)
- const int DIODE_DROP = 700

  BATTERY.
- float res3 = 2.5
- float res2 = 1.
- float totalDrop = res2 / (res2 + res3)
- uint8_t adcBits = 12
- float maximumWidth = pow(2., (float)adcBits)-1
- double fromVtoWidth = maximumWidth / (double)BOARD_LIMIT_VOLTAGE
- double maxBatteryLevelDropped = (double)(MAX_BATTERY_VOLTAGE-DIODE_DROP) * totalDrop
- double correctionBattery = (double)BOARD_LIMIT_VOLTAGE/maxBatteryLevelDropped
- double minBatteryLevelThreshold = ((double)MIN_BATTERY_VOLTAGE-(double)DIODE_DROP) * totalDrop * correctionBattery
- const float PRESSURE_SEA_LEVEL = 1013.25

  ALTIMETER.
- uint8_t osrs_t = 1
- uint8_t osrs_p = 1
- uint8_t barometerMode = 3
- uint8_t t_sb = 5
- uint8_t filter = 0
- uint8_t spi3w_en = 0
- const char * ssid = "DroneInoTelemetry"

  TELEMETRY.
- const char * password = "DroneIno"
- const int refreshRate = 200
- int refreshCounter = 0
- const char * P_ROLL_GET = "rollP"
- const char * I_ROLL_GET = "rollI"
- const char * D_ROLL_GET = "rollD"
- const char * P_YAW_GET = "yawP"
- const char * I_YAW_GET = "yawI"
- const char * D_YAW_GET = "yawD"
- const char * FILTER_P_R_GET = "filterPitchRoll"
- const char * ROLL_CORR_GET = "correctionRoll"
- const char * PITCH_CORR_GET = "correctionPitch"
- const char * P_ALTITUDE_GET = "altitudeP"
- const char * I_ALTITUDE_GET = "altitudeI"
- const char * D_ALTITUDE_GET = "altitudeD"
- const int dataTransferSize = 5

- float dataTransfer [dataTransferSize]
- const int dataControllerSize = 12
- float dataController [dataControllerSize]
- struct trimPosition trimCh [6]
- byte flightMode
- byte eepromData [36]
- byte errWire
- int16_t calInt
- int16_t start
- int error
- volatile int receiverInputChannel1
- volatile int receiverInputChannel2
- volatile int receiverInputChannel3
- volatile int receiverInputChannel4
- volatile int receiverInputChannel5
- byte lastChannel1
- byte lastChannel2
- byte lastChannel3
- byte lastChannel4
- byte lastChannel5
- unsigned long timer1
- unsigned long timer2
- unsigned long timer3
- unsigned long timer4
- unsigned long timer5
- unsigned long currentTime
- unsigned long loopTimer
- int16_t esc1
- int16_t esc2
- int16_t esc3
- int16_t esc4
- int16_t throttle
- float pidIMemRoll
- float pidRollSetpoint
- float gyroRollInput
- float pidOutputRoll
- float pidLastRollDError
- float pidIMemPitch
- float pidPitchSetpoint
- float gyroPitchInput
- float pidOutputPitch
- float pidLastPitchDError
- float pidIMemYaw
- float pidYawSetpoint
- float gyroYawInput
- float pidOutputYaw
- float pidLastYawDError
- float pidErrorTemp
- float pidErrorGainAltitude
- float pidIMemAltitude
- float pidAltitudeSetpoint
- float pidAltitudeInput
- float pidOutputAltitude
- uint8_t parachuteRotatingMemLocation
- int32_t parachuteBuffer [35]

- int32_t parachuteThrottle
- float pressureParachutePrevious
- int32_t pressureRotatingMem [50]
- int32_t pressureTotalAvarage
- uint8_t pressureRotatingMemLocation
- uint8_t manualAltitudeChange
- int16_t manualThrottle
- boolean gyroAnglesSet
- int16_t gyroTemp
- int16_t accAxis [4]
- int16_t gyroAxis [4]
- double gyroAxisCalibration [4]
- double accAxisCalibration [4]
- float angleRollAcc
- float anglePitchAcc
- float anglePitch
- float angleRoll
- float rollLevelAdjust
- float pitchLevelAdjust
- long accTotalVector
- float batteryVoltage
- float batteryPercent
- float pressure
- float altitudeMeasure
- float temperature
- uint8_t barometerCounter
- float actualPressure
- float actualPressureSlow
- float actualPressureFast
- float actualPressureDiff
- uint16_t dig_P1
- uint16_t dig_T1
- int16_t dig_T2
- int16_t dig_T3
- int16_t dig_P2
- int16_t dig_P3
- int16_t dig_P4
- int16_t dig_P5
- int16_t dig_P6
- int16_t dig_P7
- int16_t dig_P8
- int16_t dig_P9
- int16_t dig_H2
- int16_t dig_H4
- int16_t dig_H5
- int8_t dig_H1
- int8_t dig_H3
- int8_t dig_H6
- long adcP
- long adcT
- unsigned long int tempRaw
- unsigned long int presRaw
- signed long int tFine
- signed long int tempCal
- unsigned long int pressCal

5.14.1 Detailed Description

Global variables used in all the project.

Author

@sebastiano123-cs

Global variables are divided here into:

- PID: roll, yaw, pitch and altitude;
- GYROSCOPE: low/high filter percentage, pitch and roll correction;
- PWM channels: for the use of the ledWrite;
- BATTERY: battery constants and calculation of the battery level;
- ALTIMETER: constants;
- TELEMETRY: data transfer and data controller size and array;
- UNDEFINED: variables not yet defined;

Version

0.1

Date

2022-02-28

Copyright

5.14.2 Variable Documentation

5.14.2.1 accAxis

```
int16_t accAxis[4]
```

5.14.2.2 accAxisCalibration

```
double accAxisCalibration[4]
```

5.14.2.3 accTotalVector

```
long accTotalVector
```

5.14.2.4 actualPressure

```
float actualPressure
```

5.14.2.5 actualPressureDiff

```
float actualPressureDiff
```

5.14.2.6 actualPressureFast

```
float actualPressureFast
```

5.14.2.7 actualPressureSlow

```
float actualPressureSlow
```

5.14.2.8 adcBits

```
uint8_t adcBits = 12
```

5.14.2.9 adcP

```
long adcP
```

5.14.2.10 adcT

```
long adcT
```

5.14.2.11 altitudeMeasure

```
float altitudeMeasure
```

5.14.2.12 anglePitch

```
float anglePitch
```

5.14.2.13 anglePitchAcc

```
float anglePitchAcc
```

5.14.2.14 anglePitchOffset

```
float anglePitchOffset = 0
```

5.14.2.15 angleRoll

```
float angleRoll
```

5.14.2.16 angleRollAcc

```
float angleRollAcc
```

5.14.2.17 angleRollOffset

```
float angleRollOffset = 0
```

5.14.2.18 barometerCounter

```
uint8_t barometerCounter
```

5.14.2.19 barometerMode

```
uint8_t barometerMode = 3
```

5.14.2.20 batteryPercent

```
float batteryPercent
```

5.14.2.21 batteryVoltage

```
float batteryVoltage
```

5.14.2.22 calInt

```
int16_t calInt
```

5.14.2.23 convDegToRad

```
float convDegToRad = 180.0 / PI
```

5.14.2.24 correctionBattery

```
double correctionBattery = (double)BOARD_LIMIT_VOLTAGE/maxBatteryLevelDropped
```

5.14.2.25 correctionPitchRoll

```
const int correctionPitchRoll = 15
```

5.14.2.26 currentTime

```
unsigned long currentTime
```

5.14.2.27 D_ALTITUDE_GET

```
const char* D_ALTITUDE_GET = "altitudeD"
```

5.14.2.28 D_ROLL_GET

```
const char* D_ROLL_GET = "rollD"
```

5.14.2.29 D_YAW_GET

```
const char* D_YAW_GET = "yawD"
```

5.14.2.30 dataController

```
float dataController[dataControllerSize]
```

5.14.2.31 dataControllerSize

```
const int dataControllerSize = 12
```

5.14.2.32 dataTransfer

```
float dataTransfer[dataTransferSize]
```

5.14.2.33 dataTransferSize

```
const int dataTransferSize = 5
```

5.14.2.34 dig_H1

```
int8_t dig_H1
```

5.14.2.35 dig_H2

```
int16_t dig_H2
```

5.14.2.36 dig_H3

```
int8_t dig_H3
```

5.14.2.37 dig_H4

```
int16_t dig_H4
```

5.14.2.38 dig_H5

```
int16_t dig_H5
```

5.14.2.39 dig_H6

```
int8_t dig_H6
```

5.14.2.40 dig_P1

```
uint16_t dig_P1
```

5.14.2.41 dig_P2

```
int16_t dig_P2
```

5.14.2.42 dig_P3

```
int16_t dig_P3
```

5.14.2.43 dig_P4

```
int16_t dig_P4
```

5.14.2.44 dig_P5

```
int16_t dig_P5
```

5.14.2.45 dig_P6

```
int16_t dig_P6
```

5.14.2.46 dig_P7

```
int16_t dig_P7
```

5.14.2.47 dig_P8

```
int16_t dig_P8
```

5.14.2.48 dig_P9

```
int16_t dig_P9
```

5.14.2.49 dig_T1

```
uint16_t dig_T1
```

5.14.2.50 dig_T2

```
int16_t dig_T2
```

5.14.2.51 dig_T3

```
int16_t dig_T3
```

5.14.2.52 DIODE_DROP

```
const int DIODE_DROP = 700
```

BATTERY.

Note

total resistance calculations, the important quantity is totalDrop

5.14.2.53 eepromData

```
byte eepromData[36]
```

5.14.2.54 error

```
int error
```

5.14.2.55 errWire

```
byte errWire
```

5.14.2.56 esc1

```
int16_t esc1
```

5.14.2.57 esc2

```
int16_t esc2
```

5.14.2.58 esc3

```
int16_t esc3
```

5.14.2.59 esc4

```
int16_t esc4
```

5.14.2.60 filter

```
uint8_t filter = 0
```

5.14.2.61 FILTER_P_R_GET

```
const char* FILTER_P_R_GET = "filterPitchRoll"
```

5.14.2.62 flightMode

```
byte flightMode
```

5.14.2.63 freq

```
const int freq = 500
```

5.14.2.64 fromVtoWidth

```
double fromVtoWidth = maximumWidth / (double)BOARD_LIMIT_VOLTAGE
```

5.14.2.65 gyroAnglesSet

```
boolean gyroAnglesSet
```

5.14.2.66 gyroAxis

```
int16_t gyroAxis[4]
```

5.14.2.67 gyroAxisCalibration

```
double gyroAxisCalibration[4]
```

5.14.2.68 gyroFrequency

```
const int gyroFrequency = 250
```

5.14.2.69 gyroPitchInput

```
float gyroPitchInput
```

5.14.2.70 gyroRollInput

```
float gyroRollInput
```

5.14.2.71 GYROSCOPE_PITCH_CORR

```
float GYROSCOPE_PITCH_CORR = -1.65
```

5.14.2.72 GYROSCOPE_PITCH_FILTER

```
float GYROSCOPE_PITCH_FILTER = GYROSCOPE_PITCH_FILTER
```

5.14.2.73 GYROSCOPE_ROLL_CORR

```
float GYROSCOPE_ROLL_CORR = -.30
```

5.14.2.74 GYROSCOPE_ROLL_FILTER

```
float GYROSCOPE_ROLL_FILTER = .9996
```

GYROSCOPE.

5.14.2.75 gyroSensibility

```
const float gyroSensibility = 65.5
```

5.14.2.76 gyroTemp

```
int16_t gyroTemp
```

5.14.2.77 gyroYawInput

```
float gyroYawInput
```

5.14.2.78 HALF_DUTY_CYCLE

```
const int HALF_DUTY_CYCLE = (int)(0.5*MAX_DUTY_CYCLE)
```

5.14.2.79 I_ALTITUDE_GET

```
const char* I_ALTITUDE_GET = "altitudeI"
```

5.14.2.80 I_ROLL_GET

```
const char* I_ROLL_GET = "rollI"
```

5.14.2.81 I_YAW_GET

```
const char* I_YAW_GET = "yawI"
```

5.14.2.82 lastChannel1

```
byte lastChannel1
```

5.14.2.83 lastChannel2

```
byte lastChannel2
```

5.14.2.84 lastChannel3

```
byte lastChannel3
```

5.14.2.85 lastChannel4

```
byte lastChannel4
```

5.14.2.86 lastChannel5

```
byte lastChannel5
```

5.14.2.87 loopTimer

```
unsigned long loopTimer
```

5.14.2.88 manualAltitudeChange

```
uint8_t manualAltitudeChange
```

5.14.2.89 manualThrottle

```
int16_t manualThrottle
```

5.14.2.90 MAX_DUTY_CYCLE

```
const int MAX_DUTY_CYCLE = (int)(pow(2, resolution) - 1)
```

5.14.2.91 maxBatteryLevelDropped

```
double maxBatteryLevelDropped = (double)(MAX_BATTERY_VOLTAGE-DIODE_DROP) * totalDrop
```

5.14.2.92 maximumWidth

```
float maximumWidth = pow(2., (float)adcBits)-1
```

5.14.2.93 minBatteryLevelThreshold

```
double minBatteryLevelThreshold = ((double)MIN_BATTERY_VOLTAGE-(double)DIODE_DROP) * totalDrop
* correctionBattery
```

5.14.2.94 osrs_p

```
uint8_t osrs_p = 1
```

5.14.2.95 osrs_t

```
uint8_t osrs_t = 1
```

5.14.2.96 P_ALTITUDE_GET

```
const char* P_ALTITUDE_GET = "altitudeP"
```

5.14.2.97 P_ROLL_GET

```
const char* P_ROLL_GET = "rollP"
```

5.14.2.98 P_YAW_GET

```
const char* P_YAW_GET = "yawP"
```

5.14.2.99 parachuteBuffer

```
int32_t parachuteBuffer[35]
```

5.14.2.100 parachuteRotatingMemLocation

```
uint8_t parachuteRotatingMemLocation
```

5.14.2.101 parachuteThrottle

```
int32_t parachuteThrottle
```

5.14.2.102 password

```
const char* password = "DroneIno"
```

5.14.2.103 PID_D_GAIN_ALTITUDE

```
float PID_D_GAIN_ALTITUDE = 0.75
```

5.14.2.104 PID_D_GAIN_PITCH

```
float PID_D_GAIN_PITCH = PID_D_GAIN_ROLL
```

5.14.2.105 PID_D_GAIN_ROLL

```
float PID_D_GAIN_ROLL = 10.0
```

5.14.2.106 PID_D_GAIN_YAW

```
float PID_D_GAIN_YAW = 0.0
```

5.14.2.107 PID_I_GAIN_ALTITUDE

```
float PID_I_GAIN_ALTITUDE = 0.3
```

5.14.2.108 PID_I_GAIN_PITCH

```
float PID_I_GAIN_PITCH = PID_I_GAIN_ROLL
```

5.14.2.109 PID_I_GAIN_ROLL

```
float PID_I_GAIN_ROLL = 0.000001
```

5.14.2.110 PID_I_GAIN_YAW

```
float PID_I_GAIN_YAW = 0.05
```

5.14.2.111 PID_MAX_ALTITUDE

```
int PID_MAX_ALTITUDE = 400
```

5.14.2.112 PID_MAX_PITCH

```
int PID_MAX_PITCH = PID_MAX_ROLL
```

5.14.2.113 PID_MAX_ROLL

```
int PID_MAX_ROLL = 400
```

5.14.2.114 PID_MAX_YAW

```
int PID_MAX_YAW = 400
```

5.14.2.115 PID_P_GAIN_ALTITUDE

```
float PID_P_GAIN_ALTITUDE = 1.4
```

5.14.2.116 PID_P_GAIN_PITCH

```
float PID_P_GAIN_PITCH = PID_P_GAIN_ROLL
```

5.14.2.117 PID_P_GAIN_ROLL

```
float PID_P_GAIN_ROLL = 1.0
```

PID parameters for roll, pitch, yaw and altitude.

Note

Roll and pitch parameters have the same values.

5.14.2.118 PID_P_GAIN_YAW

```
float PID_P_GAIN_YAW = 0.7
```

5.14.2.119 pidAltitudeInput

```
float pidAltitudeInput
```

5.14.2.120 pidAltitudeSetpoint

```
float pidAltitudeSetpoint
```

5.14.2.121 pidErrorGainAltitude

```
float pidErrorGainAltitude
```

5.14.2.122 pidErrorTemp

```
float pidErrorTemp
```

5.14.2.123 pidIMemAltitude

```
float pidIMemAltitude
```

5.14.2.124 pidIMemPitch

```
float pidIMemPitch
```

5.14.2.125 pidIMemRoll

```
float pidIMemRoll
```

5.14.2.126 pidIMemYaw

```
float pidIMemYaw
```

5.14.2.127 pidLastPitchDError

```
float pidLastPitchDError
```

5.14.2.128 pidLastRollDError

```
float pidLastRollDError
```

5.14.2.129 pidLastYawDError

```
float pidLastYawDError
```

5.14.2.130 pidOutputAltitude

```
float pidOutputAltitude
```

5.14.2.131 pidOutputPitch

```
float pidOutputPitch
```

5.14.2.132 pidOutputRoll

```
float pidOutputRoll
```

5.14.2.133 pidOutputYaw

```
float pidOutputYaw
```

5.14.2.134 pidPitchSetpoint

```
float pidPitchSetpoint
```

5.14.2.135 pidRollSetpoint

```
float pidRollSetpoint
```

5.14.2.136 pidYawSetpoint

```
float pidYawSetpoint
```

5.14.2.137 PITCH_CORR_GET

```
const char* PITCH_CORR_GET = "correctionPitch"
```

5.14.2.138 pitchLevelAdjust

```
float pitchLevelAdjust
```

5.14.2.139 presRaw

```
unsigned long int presRaw
```

5.14.2.140 pressCal

```
unsigned long int pressCal
```

5.14.2.141 pressure

```
float pressure
```

5.14.2.142 PRESSURE_SEA_LEVEL

```
const float PRESSURE_SEA_LEVEL = 1013.25
```

ALTIMETER.

5.14.2.143 pressureParachutePrevious

```
float pressureParachutePrevious
```

5.14.2.144 pressureRotatingMem

```
int32_t pressureRotatingMem[50]
```

5.14.2.145 pressureRotatingMemLocation

```
uint8_t pressureRotatingMemLocation
```

5.14.2.146 pressureTotalAvarage

```
int32_t pressureTotalAvarage
```

5.14.2.147 pwmChannel1

```
const int pwmChannel1 = 1
```

5.14.2.148 pwmChannel2

```
const int pwmChannel2 = 2
```

5.14.2.149 pwmChannel3

```
const int pwmChannel3 = 3
```

5.14.2.150 pwmChannel4

```
const int pwmChannel4 = 4
```

5.14.2.151 pwmLedChannel

```
const int pwmLedChannel = 0
```

PWM.

Note

PWM is used for ESC pulses and LEDs

5.14.2.152 pwmLedFlyChannel

```
const int pwmLedFlyChannel = 5
```

5.14.2.153 receiverInputChannel1

```
volatile int receiverInputChannel1
```

5.14.2.154 receiverInputChannel2

```
volatile int receiverInputChannel2
```

5.14.2.155 receiverInputChannel3

```
volatile int receiverInputChannel3
```

5.14.2.156 receiverInputChannel4

```
volatile int receiverInputChannel4
```

5.14.2.157 receiverInputChannel5

```
volatile int receiverInputChannel5
```

5.14.2.158 refreshCounter

```
int refreshCounter = 0
```

5.14.2.159 refreshRate

```
const int refreshRate = 200
```

5.14.2.160 res2

```
float res2 = 1.
```

5.14.2.161 res3

```
float res3 = 2.5
```

5.14.2.162 resolution

```
const int resolution = 11
```

5.14.2.163 ROLL_CORR_GET

```
const char* ROLL_CORR_GET = "correctionRoll"
```

5.14.2.164 rollLevelAdjust

```
float rollLevelAdjust
```

5.14.2.165 spi3w_en

```
uint8_t spi3w_en = 0
```

5.14.2.166 ssid

```
const char* ssid = "DroneInoTelemetry"
```

TELEMETRY.

5.14.2.167 start

```
int16_t start
```

5.14.2.168 t_sb

```
uint8_t t_sb = 5
```

5.14.2.169 tempCal

```
signed long int tempCal
```

5.14.2.170 temperature

```
float temperature
```

5.14.2.171 tempRaw

```
unsigned long int tempRaw
```

5.14.2.172 tFine

```
signed long int tFine
```

5.14.2.173 throttle

```
int16_t throttle
```

5.14.2.174 timer1

```
unsigned long timer1
```

5.14.2.175 timer2

```
unsigned long timer2
```

5.14.2.176 timer3

```
unsigned long timer3
```

5.14.2.177 timer4

```
unsigned long timer4
```

5.14.2.178 timer5

```
unsigned long timer5
```

5.14.2.179 totalDrop

```
float totalDrop = res2 / (res2 + res3)
```

5.14.2.180 travelCoeff

```
float travelCoeff = 1/((float)gyroFrequency * gyroSensibility)
```

5.14.2.181 travelCoeffToRad

```
float travelCoeffToRad = travelCoeff / convDegToRad
```

5.14.2.182 trimCh

```
struct trimPosition trimCh[6]
```

5.15 Globals.h

Go to the documentation of this file.

```
1
30 //              (ROLL)
31 float PID_P_GAIN_ROLL            = 1.0;                  //Gain setting for the roll P-controller
       (1.3)
32 float PID_I_GAIN_ROLL            = 0.000001;             //Gain setting for the roll I-controller
       (0.0002)
33 float PID_D_GAIN_ROLL            = 10.0;                 //Gain setting for the roll D-controller
       (10.0)
34 int PID_MAX_ROLL                 = 400;                  //Maximum output of the PID-controller
       (+/-)
35
36 //              (PITCH)
37 float PID_P_GAIN_PITCH           = PID_P_GAIN_ROLL;      //Gain setting for the pitch P-controller
38 float PID_I_GAIN_PITCH           = PID_I_GAIN_ROLL;      //Gain setting for the pitch I-controller
39 float PID_D_GAIN_PITCH           = PID_D_GAIN_ROLL;      //Gain setting for the pitch D-controller
40 int PID_MAX_PITCH                = PID_MAX_ROLL;         //Maximum output of the PID-controller
       (+/-)
41
42 //              (YAW)
43 float PID_P_GAIN_YAW             = 0.7;                  //Gain setting for the pitch P-controller.
       (2.0)
44 float PID_I_GAIN_YAW             = 0.05;                 //Gain setting for the pitch I-controller.
       (0.04)
45 float PID_D_GAIN_YAW             = 0.0;                  //Gain setting for the pitch D-controller.
       (0.0)
46 int PID_MAX_YAW                  = 400;                  //Maximum output of the PID-controller
       (+/-)
47
48 //            (ALTITUDE)
49 float PID_P_GAIN_ALTITUDE        = 1.4;                  //Gain setting for the altitude P-controller
       (default = 1.4).
50 float PID_I_GAIN_ALTITUDE        = 0.3;                  //Gain setting for the altitude I-controller
       (default = 0.2).
51 float PID_D_GAIN_ALTITUDE        = 0.75;                 //Gain setting for the altitude D-controller
       (default = 0.75).
52 int PID_MAX_ALTITUDE             = 400;                  //Maximum output of the PID-controller
       (+/-).
53
```

```
54
60 //             (FILTER)
61 float GYROSCOPE_ROLL_FILTER      = .9996;                      // read your gyroscope data after the
       calibration, try different values and choose the best one
62 float GYROSCOPE_PITCH_FILTER     = GYROSCOPE_PITCH_FILTER;     // read your gyroscope data after the
       calibration, try different values and choose the best one
63 float GYROSCOPE_ROLL_CORR        = -.30;                       // (0.) after set GYROSCOPE_ROLL_FILTER,
       put here the angle roll you read eneabling DEBUG
64 float GYROSCOPE_PITCH_CORR       = -1.65;                      // (-1.65.) after set
       GYROSCOPE_PITCH_FILTER, put here the angle pitch you read eneabling DEBUG
65
66 //            (SPECIFICS)
67 const int gyroFrequency          = 250;                                  // (Hz)
68 const float gyroSensibility      = 65.5;
69 const int correctionPitchRoll    = 15;                                   // correction for the pitch and
       roll
70 float convDegToRad               = 180.0 / PI;
71 float travelCoeff                = 1/((float)gyroFrequency * gyroSensibility);
72 float travelCoeffToRad           = travelCoeff / convDegToRad;
73 float anglePitchOffset           = 0;                                    // NOT touch, for future
74 float angleRollOffset            = 0;                                    // NOT touch, for future
75
76
84 //              (CHANNELS)
85 const int pwmLedChannel          = 0;
86 const int pwmChannel1            = 1;
87 const int pwmChannel2            = 2;
88 const int pwmChannel3            = 3;
89 const int pwmChannel4            = 4;
90 const int pwmLedFlyChannel       = 5;
91
92 //              (SPECIFICS)
93 const int freq                   = 500;                                      // (Hz) for what I know, 500 is
       the best
94 const int resolution             = 11;                                       // (bits) 11 is the best guess
95 const int MAX_DUTY_CYCLE         = (int)(pow(2, resolution) - 1);
96 const int HALF_DUTY_CYCLE        = (int)(0.5*MAX_DUTY_CYCLE);
97
98
106 //              (CIRCUIT SPECIFICS)
107 const int DIODE_DROP             = 700;                                  //generally it is -0.7V
108 float res3                       = 2.5;                                  // resistance between Vin and
       vout
109 float res2                       = 1.;                                   // load resistance
110 float totalDrop                  = res2 / (res2 + res3);                 // IMPORTANT: this is in my
       case, you have to calculate YOUR total drop
111
112 //              (DIGITAL PINS ACCURACY)
113 uint8_t adcBits                  = 12;                                   // (bits) of width when
       measuring the voltage
114 float maximumWidth               = pow(2., (float)adcBits)-1;            // maximum width that the pin
       can read
115
116 //              (CALCULATIONS)
117 double fromVtoWidth              = maximumWidth / (double)BOARD_LIMIT_VOLTAGE;
118 double maxBatteryLevelDropped    = (double)(MAX_BATTERY_VOLTAGE-DIODE_DROP) * totalDrop;
119 double correctionBattery         = (double)BOARD_LIMIT_VOLTAGE/maxBatteryLevelDropped;
120 double minBatteryLevelThreshold  = ((double)MIN_BATTERY_VOLTAGE-(double)DIODE_DROP) * totalDrop *
       correctionBattery;
121
122
127 const float PRESSURE_SEA_LEVEL   = 1013.25;                       // hPa
128 uint8_t osrs_t                   = 1;                            // Temperature oversampling x 1
129 uint8_t osrs_p                   = 1;                            // Pressure oversampling x 1
130 uint8_t barometerMode            = 3;                            // Normal barometerMode
131 uint8_t t_sb                     = 5;                            // standby 1000ms
132 uint8_t filter                   = 0;                            // Filter off
133 uint8_t spi3w_en                 = 0;                            // 3-wire SPI Disable
134
140 //          (WIFI AP)
141 const char *ssid                 = "DroneInoTelemetry";
142 const char *password             = "DroneIno";
143 const int refreshRate            = 200;                                  // (ms) the refresh rate of
       the page
144 int refreshCounter               = 0;
145
146 //          (ESP32 BROWSER TAGS)
147 const char* P_ROLL_GET           = "rollP";
148 const char* I_ROLL_GET           = "rollI";
149 const char* D_ROLL_GET           = "rollD";
150 const char* P_YAW_GET            = "yawP";
151 const char* I_YAW_GET            = "yawI";
152 const char* D_YAW_GET            = "yawD";
153 const char* FILTER_P_R_GET       = "filterPitchRoll";
154 const char* ROLL_CORR_GET        = "correctionRoll";
155 const char* PITCH_CORR_GET       = "correctionPitch";
156 const char* P_ALTITUDE_GET       = "altitudeP";
```

```
157 const char* I_ALTITUDE_GET       = "altitudeI";
158 const char* D_ALTITUDE_GET       = "altitudeD";
159
160 //           (TX)
161 const int dataTransferSize       = 5;
162 float dataTransfer[dataTransferSize];
163
164 //            (RX)
165 const int dataControllerSize     = 12;
166 float dataController[dataControllerSize];
167
168
169
170 // // esp-now telemetry
171 // uint8_t broadcastAddress[] = {0x58, 0xBF, 0x25, 0x82, 0x4D, 0x08}; // RECEIVER MAC Address
172 // // Create an Event Source on /events
173 // AsyncEventSource events("/events");
174
175 // trim position structure
176 struct trimPosition{
177   byte reverse;                                          // if is inverted
178   int16_t low;                                           // low value for the specific receiver input
       channel
179   int16_t center;                                        // center value for the specific receiver
       input channel
180   int16_t high;                                          // high value for the specific receiver input
       channel
181   int16_t actual;                                        // instantaneous receiver value
182 } trimCh[6];
183
184 // flight mode
185 byte flightMode;                                         // 1 = only auto leveling (or nothing if
      AUTO_LEVELING = false), 2 = altitude hold
186
187 // very global
188 byte eepromData[36], errWire;
189 int16_t calInt, start;
190 int error;
191
192 // ISR
193 volatile int receiverInputChannel1, receiverInputChannel2, receiverInputChannel3, receiverInputChannel4,
      receiverInputChannel5;
194 byte lastChannel1, lastChannel2, lastChannel3, lastChannel4, lastChannel5;
195 unsigned long timer1, timer2, timer3, timer4, timer5, currentTime, loopTimer;
196 int16_t esc1, esc2, esc3, esc4;
197 int16_t throttle;
198
199 // PID
200 float pidIMemRoll, pidRollSetpoint, gyroRollInput, pidOutputRoll, pidLastRollDError;
201 float pidIMemPitch, pidPitchSetpoint, gyroPitchInput, pidOutputPitch, pidLastPitchDError;
202 float pidIMemYaw, pidYawSetpoint, gyroYawInput, pidOutputYaw, pidLastYawDError;
203 float pidErrorTemp;
204
205 // Altitude PID variables
206 float pidErrorGainAltitude;
207 float pidIMemAltitude, pidAltitudeSetpoint, pidAltitudeInput, pidOutputAltitude;
208 uint8_t parachuteRotatingMemLocation;
209 int32_t parachuteBuffer[35], parachuteThrottle;
210 float pressureParachutePrevious;
211 int32_t pressureRotatingMem[50], pressureTotalAvarage;
212 uint8_t pressureRotatingMemLocation;
213 uint8_t manualAltitudeChange;
214 int16_t manualThrottle;
215
216 // gyroscope
217 boolean gyroAnglesSet;
218 int16_t gyroTemp, accAxis[4], gyroAxis[4];
219 double gyroAxisCalibration[4], accAxisCalibration[4];
220 float angleRollAcc, anglePitchAcc, anglePitch, angleRoll;
221 float rollLevelAdjust, pitchLevelAdjust;
222 long accTotalVector;
223
224 // battery
225 float batteryVoltage, batteryPercent;
226
227 // altitude sensor
228 float pressure, altitudeMeasure;
229 float temperature;
230 uint8_t barometerCounter;
231 float actualPressure, actualPressureSlow, actualPressureFast, actualPressureDiff;
232
233 // BMP280 Wire.read() values
234 uint16_t dig_P1, dig_T1;
235 int16_t dig_T2, dig_T3, dig_P2, dig_P3, dig_P4, dig_P5, dig_P6, dig_P7, dig_P8, dig_P9, dig_H2, dig_H4,
      dig_H5;
236 int8_t dig_H1, dig_H3, dig_H6;
237 long adcP, adcT;
```

```
243 unsigned long int tempRaw, presRaw;
244 signed long int tFine;
245 signed long int tempCal;
246 unsigned long int pressCal;
```

5.16 DroneIno/Gyroscope.ino File Reference

Gyroscopes routines.

Functions

- void setGyroscopeRegisters ()

  Try a first communication with the gyroscope.
- void calibrateGyroscope ()

  Uses the gyroscope measurement to calculate the accelerometer and the gyroscope calibration.
- void calculateAnglePRY ()

  Calculate the pitch, roll and yaw angles from the previous gyroscope readings.
- void printGyroscopeStatus ()

  Prints gyroscope readings.
- void readGyroscopeStatus ()

  Read the gyroscope data.

5.16.1 Detailed Description

Gyroscopes routines.

Author

@sebastiano123-c

Version

0.1

Date

2022-02-28

Copyright

Copyright (c) 2022

5.16.2 Function Documentation

5.16.2.1 calculateAnglePRY()

```
void calculateAnglePRY ( )
```

Calculate the pitch, roll and yaw angles from the previous gyroscope readings.

5.16.2.2 calibrateGyroscope()

```
void calibrateGyroscope ( )
```

Uses the gyroscope measurement to calculate the accelerometer and the gyroscope calibration.

5.16.2.3 printGyroscopeStatus()

```
void printGyroscopeStatus ( )
```

Prints gyroscope readings.

5.16.2.4 readGyroscopeStatus()

```
void readGyroscopeStatus ( )
```

Read the gyroscope data.

5.16.2.5 setGyroscopeRegisters()

```
void setGyroscopeRegisters ( )
```

Try a first communication with the gyroscope.

5.17 DroneIno/Initialize.ino File Reference

Some routines used in the setup().

Functions

- void initEEPROM ()

  Initilizes the EEPROM.
- void configureReceiverTrims ()

  Fills the trims values using the EEPROM saved values.
- void setupWireI2C ()

  Setup the WIRE communication.
- void setupPins ()

  Definition of all the pinModes.
- void intro ()

  Fancy introduction with logo and author.
- void printEEPROM ()

  Print all the EEPROM data.

5.17.1 Detailed Description

Some routines used in the setup().

Author

@sebastiano123-c

Version

0.1

Date

2022-03-01

Copyright

5.17.2 Function Documentation

5.17.2.1 configureReceiverTrims()

```
void configureReceiverTrims ( )
```

Fills the trims values using the EEPROM saved values.

5.17.2.2 initEEPROM()

```
void initEEPROM ( )
```

Initilizes the EEPROM.

5.17.2.3 intro()

```
void intro ( )
```

Fancy introduction with logo and author.

5.17.2.4 printEEPROM()

```
void printEEPROM ( )
```

Print all the EEPROM data.

5.17.2.5 setupPins()

```
void setupPins ( )
```

Definition of all the pinModes.

5.17.2.6 setupWireI2C()

```
void setupWireI2C ( )
```

Setup the WIRE communication.

5.18 DroneIno/ISR.ino File Reference

ISR routine when the receiver signal is triggered.

Functions

- void myISR ()

  Measures the receiver input signal length.

5.18.1 Detailed Description

ISR routine when the receiver signal is triggered.

Author

@sebastiano123-c

Version

0.1

Date

2022-03-01

Copyright

Copyright (c) 2022

5.18.2 Function Documentation

5.18.2.1 myISR()

```
void myISR ( )
```

Measures the receiver input signal length.

Todo Use pwm to read the receivers signals, avoid digitalRead.

5.19 DroneIno/PID.ino File Reference

PID routines.

Functions

- void setPID ()

  Set the PID adjustments for output.
- void calculatePID ()

  Calculates PID adjustments for outputs. Calculates PID in degrees per seconds.

5.19.1 Detailed Description

PID routines.

Author

@sebastiano123-c

Version

0.1

Date

2022-02-28

Copyright

5.19.2 Function Documentation

5.19.2.1 calculatePID()

```
void calculatePID ( )
```

Calculates PID adjustments for outputs. Calculates PID in degrees per seconds.

5.19.2.2 setPID()

```
void setPID ( )
```

Set the PID adjustments for output.

Note

PID set point in degrees per second is determined by the roll receiver input.

5.20 DroneIno/WiFi.ino File Reference

WiFi network routines for telemetry and PID settingusing web apps.

Functions

- void notFound (AsyncWebServerRequest *request)

  Not found response.
- const char * index_html ()

  HTML source of the web page.
- String processor (const String &var)

  Substittue the HTML tag with the variable.
- void setupWiFiTelemetry ()

  Setup the wifi server AP (generated by the ESP32).
- void sendWiFiTelemetry ()

  Updates '\events' of the web server with the sensor readings.

5.20.1 Detailed Description

WiFi network routines for telemetry and PID settingusing web apps.

Author

your name (you@domain.com)

Define in the Config.h file the macro WIFI_TELEMETRY as:

- NATIVE: if you don't have a ESP32-CAM. This will use the ESP32 native wifi to easily set PID parameters;
- ESP_CAM: allow you to use an ESP32-CAM mounting a OV2640 camera as a telemetry system.

NATIVE it is not actually a proper telemetry system since it allows only to change PID very easily and on the fly, but not to see telemetry data. It is so because the ping time between the message sent and the received positive response is an eternity (like 10ms) with respect to the 4ms rate.

ESP_CAM, connected to DroneIno accordingly to the documentation on GitHub, provides a full telemetry system, a video streaming and the possibility to set PID parameters back to DroneIno.

Both uses, network name "DroneInoTelemetry" passwork "DroneIno" server IP "192.168.4.1"

After connecting to DroneInoTelemetry network, dial "192.168.4.1" on your browser.

Version

0.1

Date

2022-03-01

Copyright

5.20.2 Function Documentation

5.20.2.1 index_html()

```
const char * index_html ( )
```

HTML source of the web page.

Returns

const char*

5.20.2.2 notFound()

```
void notFound (
            AsyncWebServerRequest * request )
```

Not found response.

Parameters

request	

5.20.2.3 processor()

```
String processor (
            const String & var )
```

Substittue the HTML tag with the variable.

Parameters

var	

Returns

String

5.20.2.4 sendWiFiTelemetry()

```
void sendWiFiTelemetry ( )
```

Updates '\events' of the web server with the sensor readings.

Note

refresh rate browser is about 20 Hertz (i.e. they refresh the screen about 20 times per second) - i.e. every 50 milliseconds

Returns

* void

5.20.2.5 setupWiFiTelemetry()

```
void setupWiFiTelemetry ( )
```

Setup the wifi server AP (generated by the ESP32).

Note

Web app link: 192.168.4.1

5.21 Setup/Setup.ino File Reference

Setup sketch to check DroneIno sensors, motors, receiver and leds.

```
#include <Arduino.h>
#include <Wire.h>
#include <EEPROM.h>
```

Macros

- #define BAUD_RATE 115200
- #define EEPROM_SIZE 36
- #define WIRE_CLOCK 400000L
- #define PIN_ESC_1 17
- #define PIN_ESC_2 16
- #define PIN_ESC_3 27
- #define PIN_ESC_4 14
- #define PIN_RECEIVER_1 12
- #define PIN_RECEIVER_2 13
- #define PIN_RECEIVER_3 5
- #define PIN_RECEIVER_4 23
- #define PIN_BATTERY_LED 19
- #define PIN_DIGITAL_13 18
- #define PIN_BATTERY_LEVEL 2
- #define PIN_PROXIMITY_SENSOR_ECHO 26
- #define PIN_PROXIMITY_SENSOR_TRIG 25
- #define PIN_ALTITUDE_SENSOR_SDA 36
- #define PIN_ALTITUDE_SENSOR_SCL 39

Functions

- void setup ()
- void loop ()
- byte search_gyro (int gyroAddress, int who_am_i)
- void start_gyro ()
- void gyro_signalen ()
- void check_receiver_inputs (byte movement)
- void check_to_continue ()
- void wait_sticks_zero ()
- void wait_for_receiver ()
- void register_min_max ()
- void check_gyro_axes (byte movement)
- void myISR ()
- void intro ()

Variables

- byte lastChannel1
- byte lastChannel2
- byte lastChannel3
- byte lastChannel4
- byte lowByte
- byte highByte
- byte type
- byte gyroAddress
- byte error
- byte clockspeedOk
- byte channel1Assign
- byte channel2Assign
- byte channel3Assign
- byte channel4Assign
- byte rollAxis
- byte pitchAxis
- byte yawAxis
- byte receiverCheckByte
- byte gyroCheckByte
- volatile int receiverInputChannel1
- volatile int receiverInputChannel2
- volatile int receiverInputChannel3
- volatile int receiverInputChannel4
- int16_t centerChannel1
- int16_t centerChannel2
- int16_t centerChannel3
- int16_t centerChannel4
- int16_t highChannel1
- int16_t highChannel2
- int16_t highChannel3
- int16_t highChannel4
- int16_t lowChannel1
- int16_t lowChannel2
- int16_t lowChannel3
- int16_t lowChannel4
- int16_t address

- int16_t calInt
- unsigned long timer
- unsigned long timer1
- unsigned long timer2
- unsigned long timer3
- unsigned long timer4
- unsigned long currentTime
- int16_t gyroPitch
- int16_t gyroRoll
- int16_t gyroYaw
- int16_t gyroRollCal
- int16_t gyroPitchCal
- int16_t gyroYawCal

5.21.1 Detailed Description

Setup sketch to check DroneIno sensors, motors, receiver and leds.

Author

@sebastiano123-c

Version

0.1

Date

2022-02-28

TERMS OF USE

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Note

Always remove the propellers and stay away from the motors unless you are 100% certain of what you are doing.

Copyright

5.21.2 Macro Definition Documentation

5.21.2.1 BAUD_RATE

```
#define BAUD_RATE 115200
```

5.21.2.2 EEPROM_SIZE

```
#define EEPROM_SIZE 36
```

5.21.2.3 PIN_ALTITUDE_SENSOR_SCL

```
#define PIN_ALTITUDE_SENSOR_SCL 39
```

5.21.2.4 PIN_ALTITUDE_SENSOR_SDA

```
#define PIN_ALTITUDE_SENSOR_SDA 36
```

5.21.2.5 PIN_BATTERY_LED

```
#define PIN_BATTERY_LED 19
```

5.21.2.6 PIN_BATTERY_LEVEL

```
#define PIN_BATTERY_LEVEL 2
```

5.21.2.7 PIN_DIGITAL_13

```
#define PIN_DIGITAL_13 18
```

5.21.2.8 PIN_ESC_1

```
#define PIN_ESC_1 17
```

5.21.2.9 PIN_ESC_2

```
#define PIN_ESC_2 16
```

5.21.2.10 PIN_ESC_3

```
#define PIN_ESC_3 27
```

5.21.2.11 PIN_ESC_4

```
#define PIN_ESC_4 14
```

5.21.2.12 PIN_PROXIMITY_SENSOR_ECHO

```
#define PIN_PROXIMITY_SENSOR_ECHO 26
```

5.21.2.13 PIN_PROXIMITY_SENSOR_TRIG

```
#define PIN_PROXIMITY_SENSOR_TRIG 25
```

5.21.2.14 PIN_RECEIVER_1

```
#define PIN_RECEIVER_1 12
```

5.21.2.15 PIN_RECEIVER_2

```
#define PIN_RECEIVER_2 13
```

5.21.2.16 PIN_RECEIVER_3

```
#define PIN_RECEIVER_3 5
```

5.21.2.17 PIN_RECEIVER_4

```
#define PIN_RECEIVER_4 23
```

5.21.2.18 WIRE_CLOCK

```
#define WIRE_CLOCK 400000L
```

5.21.3 Function Documentation

5.21.3.1 check_gyro_axes()

```
void check_gyro_axes (
            byte movement )
```

5.21.3.2 check_receiver_inputs()

```
void check_receiver_inputs (
            byte movement )
```

5.21.3.3 check_to_continue()

```
void check_to_continue ( )
```

5.21.3.4 gyro_signalen()

```
void gyro_signalen ( )
```

5.21.3.5 intro()

```
void intro ( )
```

5.21.3.6 loop()

```
void loop ( )
```

5.21.3.7 myISR()

```
void myISR ( )
```

5.21.3.8 register_min_max()

```
void register_min_max ( )
```

5.21.3.9 search_gyro()

```
byte search_gyro (
            int gyroAddress,
            int who_am_i )
```

5.21.3.10 setup()

```
void setup ( )
```

5.21.3.11 start_gyro()

```
void start_gyro ( )
```

5.21.3.12 wait_for_receiver()

```
void wait_for_receiver ( )
```

5.21.3.13 wait_sticks_zero()

```
void wait_sticks_zero ( )
```

5.21.4 Variable Documentation

5.21.4.1 address

```
int16_t address
```

5.21.4.2 calInt

```
int16_t calInt
```

5.21.4.3 centerChannel1

```
int16_t centerChannel1
```

5.21.4.4 centerChannel2

```
int16_t centerChannel2
```

5.21.4.5 centerChannel3

```
int16_t centerChannel3
```

5.21.4.6 centerChannel4

```
int16_t centerChannel4
```

5.21.4.7 channel1Assign

```
byte channel1Assign
```

5.21.4.8 channel2Assign

```
byte channel2Assign
```

5.21.4.9 channel3Assign

```
byte channel3Assign
```

5.21.4.10 channel4Assign

```
byte channel4Assign
```

5.21.4.11 clockspeedOk

```
byte clockspeedOk
```

5.21.4.12 currentTime

```
unsigned long currentTime
```

5.21.4.13 error

```
byte error
```

5.21.4.14 gyroAddress

```
byte gyroAddress
```

5.21.4.15 gyroCheckByte

```
byte gyroCheckByte
```

5.21.4.16 gyroPitch

```
int16_t gyroPitch
```

5.21.4.17 gyroPitchCal

```
int16_t gyroPitchCal
```

5.21.4.18 gyroRoll

```
int16_t gyroRoll
```

5.21.4.19 gyroRollCal

```
int16_t gyroRollCal
```

5.21.4.20 gyroYaw

```
int16_t gyroYaw
```

5.21.4.21 gyroYawCal

```
int16_t gyroYawCal
```

5.21.4.22 highByte

```
byte highByte
```

5.21.4.23 highChannel1

```
int16_t highChannel1
```

5.21.4.24 highChannel2

```
int16_t highChannel2
```

5.21.4.25 highChannel3

```
int16_t highChannel3
```

5.21.4.26 highChannel4

```
int16_t highChannel4
```

5.21.4.27 lastChannel1

```
byte lastChannel1
```

5.21.4.28 lastChannel2

```
byte lastChannel2
```

5.21.4.29 lastChannel3

```
byte lastChannel3
```

5.21.4.30 lastChannel4

```
byte lastChannel4
```

5.21.4.31 lowByte

```
byte lowByte
```

5.21.4.32 lowChannel1

```
int16_t lowChannel1
```

5.21.4.33 lowChannel2

```
int16_t lowChannel2
```

5.21.4.34 lowChannel3

```
int16_t lowChannel3
```

5.21.4.35 lowChannel4

```
int16_t lowChannel4
```

5.21.4.36 pitchAxis

```
byte pitchAxis
```

5.21.4.37 receiverCheckByte

```
byte receiverCheckByte
```

5.21.4.38 receiverInputChannel1

```
volatile int receiverInputChannel1
```

5.21.4.39 receiverInputChannel2

```
volatile int receiverInputChannel2
```

5.21.4.40 receiverInputChannel3

```
volatile int receiverInputChannel3
```

5.21.4.41 receiverInputChannel4

```
volatile int receiverInputChannel4
```

5.21.4.42 rollAxis

```
byte rollAxis
```

5.21.4.43 timer

```
unsigned long timer
```

5.21.4.44 timer1

```
unsigned long timer1
```

5.21.4.45 timer2

```
unsigned long timer2
```

5.21.4.46 timer3

```
unsigned long timer3
```

5.21.4.47 timer4

```
unsigned long timer4
```

5.21.4.48 type

```
byte type
```

5.21.4.49 yawAxis

```
byte yawAxis
```

5.22 WiFiTelemetry/app_httpd.cpp File Reference

Defines the routines for the communication between client (web app) and the server (esp32 cam).

```
#include "app_httpd.h"
```

Classes

- struct ra_filter_t
- struct jpg_chunking_t

Macros

- #define ENROLL_CONFIRM_TIMES 5
- #define FACE_ID_SAVE_NUMBER 7
- #define FACE_COLOR_WHITE 0x00FFFFFF
- #define FACE_COLOR_BLACK 0x00000000
- #define FACE_COLOR_RED 0x000000FF
- #define FACE_COLOR_GREEN 0x0000FF00
- #define FACE_COLOR_BLUE 0x00FF0000
- #define FACE_COLOR_YELLOW (FACE_COLOR_RED | FACE_COLOR_GREEN)
- #define FACE_COLOR_CYAN (FACE_COLOR_BLUE | FACE_COLOR_GREEN)
- #define FACE_COLOR_PURPLE (FACE_COLOR_BLUE | FACE_COLOR_RED)
- #define PART_BOUNDARY "123456789000000000000987654321"

Functions

- void startCameraServer ()

  Main routine governing the camera server.

Variables

- httpd_handle_t stream_httpd = NULL
- httpd_handle_t camera_httpd = NULL

5.22.1 Detailed Description

Defines the routines for the communication between client (web app) and the server (esp32 cam).

Author

@sebastiano123-c

Version

0.1

Date

2022-03-01

Copyright

5.22.2 Macro Definition Documentation

5.22.2.1 ENROLL_CONFIRM_TIMES

```
#define ENROLL_CONFIRM_TIMES 5
```

5.22.2.2 FACE_COLOR_BLACK

```
#define FACE_COLOR_BLACK 0x00000000
```

5.22.2.3 FACE_COLOR_BLUE

```
#define FACE_COLOR_BLUE 0x00FF0000
```

5.22.2.4 FACE_COLOR_CYAN

```
#define FACE_COLOR_CYAN (FACE_COLOR_BLUE | FACE_COLOR_GREEN)
```

5.22.2.5 FACE_COLOR_GREEN

```
#define FACE_COLOR_GREEN 0x0000FF00
```

5.22.2.6 FACE_COLOR_PURPLE

```
#define FACE_COLOR_PURPLE (FACE_COLOR_BLUE | FACE_COLOR_RED)
```

5.22.2.7 FACE_COLOR_RED

```
#define FACE_COLOR_RED 0x000000FF
```

5.22.2.8 FACE_COLOR_WHITE

```
#define FACE_COLOR_WHITE 0x00FFFFFF
```

5.22.2.9 FACE_COLOR_YELLOW

```
#define FACE_COLOR_YELLOW (FACE_COLOR_RED | FACE_COLOR_GREEN)
```

5.22.2.10 FACE_ID_SAVE_NUMBER

```
#define FACE_ID_SAVE_NUMBER 7
```

5.22.2.11 PART_BOUNDARY

```
#define PART_BOUNDARY "12345678900000000000987654321"
```

5.22.3 Function Documentation

5.22.3.1 startCameraServer()

```
void startCameraServer ( )
```

Main routine governing the camera server.

5.22.4 Variable Documentation

5.22.4.1 camera_httpd

```
httpd_handle_t camera_httpd = NULL
```

5.22.4.2 stream_httpd

```
httpd_handle_t stream_httpd = NULL
```

5.23 WiFiTelemetry/app_httpd.h File Reference

```
#include "esp_http_server.h"
#include "esp_timer.h"
#include "esp_camera.h"
#include "img_converters.h"
#include "camera_index.h"
#include "Arduino.h"
#include <stdarg.h>
#include <stdio.h>
#include "fb_gfx.h"
#include "fd_forward.h"
#include "fr_forward.h"
#include "telemetry.h"
#include "camSD.h"
```

Functions

- void startCameraServer ()

  Main routine governing the camera server.

5.23.1 Function Documentation

5.23.1.1 startCameraServer()

```
void startCameraServer ( )
```

Main routine governing the camera server.

5.24 app_httpd.h

Go to the documentation of this file.

```
1
2 #pragma once
3
4 #include "esp_http_server.h"
5 #include "esp_timer.h"
6 #include "esp_camera.h"
7 #include "img_converters.h"
8 #include "camera_index.h"
9 #include "Arduino.h"
10
11 #include <stdarg.h>
12 #include <stdio.h>
13
14 #include "fb_gfx.h"
15 #include "fd_forward.h"
16 #include "fr_forward.h"
17
18 #include "telemetry.h"
19 #include "camSD.h"
20
21 void startCameraServer();
```

5.25 WiFiTelemetry/camera_index.h File Reference

Index of the HTML source.

Macros

- #define index_ov2640_html_gz_len 4316
- #define DEFAULT 0
- #define USER_DEFINED 1
- #define CODE USER_DEFINED
- #define index_ov3660_html_gz_len 4408

Variables

- const uint8_t index_ov2640_html_gz [ ]
- const uint8_t index_ov3660_html_gz [ ]

5.25.1 Detailed Description

Index of the HTML source.

Author

@sebastiano123-c

HTML.index:

- DEFAULT: the default index, the one that you find the example sketch;
- USER_DEFINED: the user defined code.

Version

0.1

Date

2022-03-01

Copyright

5.25.2 Macro Definition Documentation

5.25.2.1 CODE

```
#define CODE USER_DEFINED
```

5.25.2.2 DEFAULT

```
#define DEFAULT 0
```

5.25.2.3 index_ov2640_html_gz_len

```
#define index_ov2640_html_gz_len 4316
```

5.25.2.4 index_ov3660_html_gz_len

```
#define index_ov3660_html_gz_len 4408
```

5.25.2.5 USER_DEFINED

```
#define USER_DEFINED 1
```

5.25.3 Variable Documentation

5.25.3.1 index_ov2640_html_gz

```
const uint8_t index_ov2640_html_gz[]
```

Initial value:

```
= {

      0x1f,0x8b,0x08,0x08,0x40,0x56,0x0f,0x62,0x00,0xff,0x69,0x6e,0x64,0x65,0x78,0x2e,0x68,0x74,0x6d,0x6c,0x2e,0x67,0x7a,0x0
}
```

5.25.3.2 index_ov3660_html_gz

```
const uint8_t index_ov3660_html_gz[]
```

5.26 camera_index.h

Go to the documentation of this file.

```
1
17 #define index_ov2640_html_gz_len  4316              // file: index_ov2640.html.gz, Size: 4316
18
19 #define DEFAULT                   0                 // original camera code
20 #define USER_DEFINED              1                 // user defined code
21
22 #define CODE                      USER_DEFINED      // (DEFAULT, USER_DEFINED)
23
24 #if CODE == USER_DEFINED    // mine html
25 const uint8_t index_ov2640_html_gz[] = {
26
      0x1f,0x8b,0x08,0x08,0x40,0x56,0x0f,0x62,0x00,0xff,0x69,0x6e,0x64,0x65,0x78,0x2e,0x68,0x74,0x6d,0x6c,0x2e,0x67,0x7a,0x0
27 };
28 #elif CODE == DEFAULT  // default camera html
29 const uint8_t index_ov2640_html_gz[] = {
30   0x1F, 0x8B, 0x08, 0x08, 0x50, 0x5C, 0xAE, 0x5C, 0x00, 0x03, 0x69, 0x6E, 0x64, 0x65, 0x78, 0x5F,
31   0x6F, 0x76, 0x32, 0x36, 0x34, 0x30, 0x2E, 0x68, 0x74, 0x6D, 0x6C, 0x00, 0xE5, 0x5D, 0x7B, 0x73,
32   0xD3, 0xC6, 0x16, 0xFF, 0x9F, 0x4F, 0x21, 0x04, 0x25, 0xF6, 0x34, 0x76, 0x6C, 0xC7, 0x84, 0xE0,
33   0xDA, 0xE2, 0x42, 0x08, 0xD0, 0x19, 0x5E, 0x25, 0x2D, 0x74, 0xA6, 0xD3, 0x81, 0xB5, 0xB4, 0xB2,
34   0x55, 0x64, 0xC9, 0x95, 0x56, 0x76, 0x52, 0x26, 0x9F, 0xE3, 0x7E, 0xA0, 0xFB, 0xC5, 0xEE, 0xD9,
35   0x87, 0xA4, 0x95, 0xBC, 0x7A, 0xD8, 0x26, 0x36, 0x97, 0xEB, 0xCC, 0x14, 0xD9, 0xDA, 0x73, 0xF6,
36   0x9C, 0xF3, 0x3B, 0xAF, 0x5D, 0x3D, 0x3A, 0xBC, 0x6D, 0xF9, 0x26, 0xB9, 0x9A, 0x63, 0x6D, 0x4A,
37   0x66, 0xAE, 0x71, 0x6B, 0xC8, 0xFF, 0xD1, 0xE0, 0x33, 0x9C, 0x62, 0x64, 0xF1, 0x43, 0xF6, 0x75,
38   0x86, 0x09, 0xD2, 0xCC, 0x29, 0x0A, 0x42, 0x4C, 0x46, 0x7A, 0x44, 0xEC, 0xD6, 0xA9, 0x9E, 0x3F,
39   0xED, 0xA1, 0x19, 0x1E, 0xE9, 0x0B, 0x07, 0x2F, 0xE7, 0x7E, 0x40, 0x74, 0xCD, 0xF4, 0x3D, 0x82,
```

```
40  0x3D, 0x18, 0xBE, 0x74, 0x2C, 0x32, 0x1D, 0x59, 0x78, 0xE1, 0x98, 0xB8, 0xC5, 0xBE, 0x1C, 0x3A,
41  0x9E, 0x43, 0x1C, 0xE4, 0xB6, 0x42, 0x13, 0xB9, 0x78, 0xD4, 0x95, 0x79, 0x11, 0x87, 0xB8, 0xD8,
42  0x38, 0xBF, 0x78, 0x7B, 0xDC, 0xD3, 0xDE, 0xBC, 0xEF, 0xF5, 0x4F, 0x3A, 0xC3, 0x23, 0xFE, 0x5B,
43  0x3A, 0x26, 0x24, 0x57, 0xF2, 0x77, 0xFA, 0x19, 0xFB, 0xD6, 0x95, 0xF6, 0x25, 0xF3, 0x13, 0xFD,
44  0xD8, 0x20, 0x44, 0xCB, 0x46, 0x33, 0xC7, 0xBD, 0x1A, 0x68, 0x8F, 0x03, 0x98, 0xF3, 0xF0, 0x05,
45  0x76, 0x17, 0x98, 0x38, 0x26, 0x3A, 0x0C, 0x91, 0x17, 0xB6, 0x42, 0x1C, 0x38, 0xF6, 0x4F, 0x2B,
46  0x84, 0x63, 0x64, 0x7E, 0x9E, 0x04, 0x7E, 0xE4, 0x59, 0x03, 0xED, 0x4E, 0xF7, 0x94, 0xFE, 0xAD,
47  0x0E, 0x32, 0x7D, 0xD7, 0x0F, 0xE0, 0xFC, 0xF9, 0x33, 0xFA, 0xB7, 0x7A, 0x9E, 0xCD, 0x1E, 0x3A,
48  0xFF, 0xE0, 0x81, 0xD6, 0x3D, 0x99, 0x5F, 0x66, 0xCE, 0x5F, 0xDF, 0xCA, 0x7C, 0x9D, 0xF6, 0x8A,
49  0xA4, 0x17, 0xF4, 0xA7, 0xE5, 0xF4, 0x21, 0x36, 0x89, 0xE3, 0x7B, 0xED, 0x19, 0x72, 0x3C, 0x05,
50  0x27, 0xCB, 0x09, 0xE7, 0x2E, 0x02, 0x1B, 0xD8, 0x2E, 0x2E, 0xE5, 0x73, 0x67, 0x86, 0xBD, 0xE8,
51  0xB0, 0x82, 0x1B, 0x65, 0xD2, 0xB2, 0x9C, 0x80, 0x8F, 0x1A, 0x50, 0x3B, 0x44, 0x33, 0xAF, 0x92,
52  0x6D, 0x99, 0x5C, 0x9E, 0xEF, 0x61, 0x85, 0x01, 0xE9, 0x44, 0xCB, 0x00, 0xCD, 0xE9, 0x00, 0xFA,
53  0xEF, 0xEA, 0x90, 0x99, 0xE3, 0x71, 0xA7, 0x1A, 0x68, 0xC7, 0xFD, 0xCE, 0xFC, 0xB2, 0x02, 0xCA,
54  0xE3, 0x13, 0xFA, 0xB7, 0x3A, 0x68, 0x8E, 0x2C, 0xCB, 0xF1, 0x26, 0x03, 0xED, 0x54, 0xC9, 0xC2,
55  0x0F, 0x2C, 0x1C, 0xB4, 0x02, 0x64, 0x39, 0x51, 0x38, 0xD0, 0xFA, 0xAA, 0x31, 0x33, 0x14, 0x4C,
56  0x40, 0x16, 0xE2, 0x83, 0xB0, 0xAD, 0xAE, 0x52, 0x12, 0x31, 0x24, 0x70, 0x26, 0x53, 0x02, 0x90,
57  0xAE, 0x8C, 0xC9, 0x1B, 0x4D, 0x84, 0x50, 0x15, 0x9E, 0xA5, 0x76, 0x53, 0x5B, 0x0D, 0xB9, 0xCE,
58  0xC4, 0x6B, 0x39, 0x04, 0xCF, 0x40, 0x9D, 0x90, 0x04, 0x98, 0x98, 0xD3, 0x32, 0x51, 0x6C, 0x67,
59  0x12, 0x05, 0x58, 0x21, 0x48, 0x62, 0xB7, 0x12, 0x85, 0xE1, 0xE4, 0xEA, 0xA9, 0xD6, 0x12, 0x8F,
60  0x3F, 0x3B, 0xA4, 0x25, 0x6C, 0x32, 0xC6, 0xB6, 0x1F, 0x60, 0xE5, 0xC8, 0x78, 0x84, 0xEB, 0x9B,
61  0x9F, 0x5B, 0x21, 0x41, 0x01, 0xA9, 0xC3, 0x10, 0xD9, 0x04, 0x07, 0xD5, 0xFC, 0x30, 0xF5, 0x8A,
62  0x6A, 0x6E, 0xC5, 0xD3, 0x8A, 0x01, 0x8E, 0xE7, 0x3A, 0x1E, 0xAE, 0x2F, 0x5E, 0xD1, 0xBC, 0x59,
63  0x76, 0x7C, 0x54, 0x0D, 0x60, 0x9C, 0xD9, 0xA4, 0xCC, 0x4B, 0x98, 0xAE, 0xAB, 0x93, 0x89, 0xB8,
64  0xE9, 0x76, 0x3A, 0x3F, 0xAC, 0x9E, 0x9C, 0x62, 0xEE, 0xA6, 0x28, 0x22, 0xFE, 0xF6, 0x11, 0xB1,
65  0x12, 0x56, 0x39, 0x3D, 0xFE, 0x35, 0xC3, 0x96, 0x83, 0xB4, 0x86, 0x14, 0xCE, 0xA7, 0x1D, 0xF0,
66  0xA9, 0xA6, 0x86, 0x3C, 0x4B, 0x6B, 0xF8, 0x81, 0x03, 0x81, 0x80, 0x58, 0xBA, 0x71, 0xE1, 0x17,
67  0x28, 0x1C, 0x73, 0xDC, 0x54, 0xA8, 0x5C, 0x12, 0x33, 0xB2, 0x45, 0xD4, 0x61, 0x43, 0x3F, 0x35,
68  0x52, 0x0E, 0xFD, 0x54, 0x06, 0x90, 0x42, 0x47, 0xC6, 0xBE, 0x0C, 0x2F, 0x59, 0xC2, 0x22, 0xCC,
69  0xE8, 0x67, 0x86, 0x2E, 0x5B, 0xA5, 0xD8, 0xC5, 0x83, 0x62, 0x0C, 0xA1, 0xCC, 0x9A, 0x0D, 0x18,
70  0xBA, 0x98, 0x6A, 0x2D, 0x8D, 0x66, 0xC9, 0xA6, 0x9A, 0x46, 0x30, 0x55, 0x43, 0x4E, 0x3F, 0xB2,
71  0x53, 0xAC, 0xA1, 0xAE, 0x5A, 0xD5, 0x34, 0x77, 0xF0, 0x3F, 0x95, 0x0F, 0x71, 0x4D, 0x0A, 0xB3,
72  0x08, 0xFD, 0xD4, 0xCF, 0x24, 0x29, 0xB3, 0xCA, 0x6C, 0xA2, 0x60, 0x5C, 0x9C, 0x51, 0x56, 0xF8,
73  0x16, 0x45, 0xB7, 0x82, 0x6B, 0xB9, 0x08, 0x75, 0xB3, 0x8B, 0x82, 0x71, 0x99, 0x0C, 0x95, 0x59,
74  0x86, 0x7E, 0xAE, 0x6B, 0xF4, 0x1B, 0x77, 0xC6, 0x11, 0x21, 0xBE, 0x17, 0x6E, 0x55, 0xA2, 0x8A,
75  0xE2, 0xEC, 0xAF, 0x28, 0x24, 0x8E, 0x7D, 0xD5, 0x12, 0x21, 0x0D, 0x71, 0x36, 0x47, 0xD0, 0x42,
76  0x8E, 0x31, 0x59, 0x62, 0x5C, 0xDE, 0x6E, 0x78, 0x68, 0x01, 0x79, 0x67, 0x32, 0x71, 0x55, 0xBE,
77  0x67, 0x46, 0x41, 0x48, 0xFB, 0xB6, 0xB9, 0xEF, 0x00, 0xE3, 0x60, 0x75, 0xE2, 0x6C, 0x0C, 0xD6,
78  0x9C, 0xA8, 0x65, 0x8E, 0x15, 0x73, 0xF9, 0x11, 0xA1, 0x36, 0x56, 0x22, 0xE1, 0x83, 0x3A, 0x0E,
79  0xB9, 0x52, 0x9E, 0x13, 0x91, 0xA8, 0x38, 0x13, 0x87, 0x60, 0x69, 0x59, 0xC8, 0xCA, 0x35, 0x30,
80  0xA7, 0xD8, 0xFC, 0x8C, 0xAD, 0x1F, 0x2B, 0xDB, 0xB0, 0xAA, 0xF6, 0xB0, 0xED, 0x78, 0xF3, 0x88,
81  0xB4, 0x68, 0x3B, 0x35, 0xBF, 0x11, 0xCC, 0x99, 0x43, 0xC6, 0x2A, 0xF6, 0x7A, 0x65, 0x4D, 0xC5,
82  0xFD, 0xF9, 0x65, 0xB9, 0x11, 0x64, 0x61, 0x0D, 0x17, 0x8D, 0xB1, 0x5B, 0x26, 0xB2, 0x08, 0x86,
83  0x82, 0xB4, 0x2B, 0x72, 0x55, 0x71, 0xEF, 0xC6, 0x24, 0x4B, 0x8B, 0x57, 0xFF, 0xC1, 0x0F, 0xB5,
84  0xED, 0xC8, 0x8E, 0x0F, 0x33, 0x3F, 0x85, 0xD8, 0x85, 0x00, 0x2B, 0x6A, 0xBD, 0x61, 0xCC, 0x12,
85  0x64, 0x28, 0x9D, 0x20, 0x40, 0xDE, 0x04, 0x43, 0x2E, 0xB8, 0x3C, 0x8C, 0x0F, 0xCB, 0x17, 0x06,
86  0xB5, 0xD4, 0xA7, 0xA9, 0xFA, 0x7E, 0xF9, 0x42, 0x84, 0x27, 0x84, 0x0D, 0x9A, 0x11, 0x09, 0xD6,
87  0xD2, 0xF9, 0xBB, 0x4A, 0xA7, 0xE0, 0xFD, 0x88, 0x32, 0x60, 0xB2, 0x2E, 0xA5, 0xEC, 0xEF, 0x2B,
88  0x33, 0x42, 0xBC, 0xD2, 0xB3, 0xED, 0xAA, 0xB5, 0xA2, 0x6D, 0x1F, 0x77, 0x8E, 0xFB, 0x95, 0x0D,
89  0x93, 0x52, 0xCB, 0xDC, 0x7A, 0x51, 0x91, 0x31, 0x92, 0x6C, 0x52, 0x0D, 0xC1, 0x60, 0xEA, 0x2F,
90  0x70, 0xA0, 0x00, 0x22, 0x27, 0x6E, 0xFF, 0x61, 0xDF, 0xAA, 0xC1, 0x0D, 0x41, 0xBE, 0x5F, 0xA8,
91  0xB2, 0x69, 0x96, 0x5D, 0xAF, 0x6B, 0xF6, 0x4A, 0x1D, 0x93, 0xB3, 0x6B, 0x83, 0x37, 0xA0, 0xB1,
92  0x8B, 0xAD, 0x92, 0xF4, 0x6C, 0x61, 0x1B, 0x45, 0x2E, 0xA9, 0xB0, 0x37, 0xEA, 0xD0, 0xBF, 0xB2,
93  0x19, 0x59, 0x5C, 0xFD, 0x41, 0x37, 0x3A, 0x46, 0x2C, 0x12, 0xFE, 0x54, 0xCC, 0x19, 0xD7, 0x4E,
94  0x34, 0x9F, 0x63, 0x04, 0xA3, 0x4C, 0x5C, 0xB4, 0x24, 0xAD, 0xD5, 0x33, 0xAB, 0x13, 0x57, 0xAD,
95  0x85, 0x68, 0xA5, 0x2B, 0x26, 0xDD, 0xD0, 0x5A, 0x3A, 0x0F, 0x6C, 0xDF, 0x8C, 0x54, 0x65, 0xBA,
96  0x9E, 0x4B, 0xAD, 0xF2, 0x1B, 0xC4, 0x26, 0x0B, 0x5D, 0x87, 0x39, 0x76, 0xE4, 0x79, 0x14, 0xD1,
97  0x16, 0x09, 0x40, 0x4D, 0xC5, 0x44, 0xF5, 0x0C, 0xB7, 0x51, 0x74, 0x66, 0x0C, 0x5B, 0xB4, 0x19,
98  0x93, 0x0B, 0x40, 0x45, 0xA2, 0x48, 0x72, 0x88, 0x16, 0xFA, 0xA0, 0x54, 0xCC, 0x6A, 0x3B, 0xBB,
99  0x90, 0x69, 0x34, 0x53, 0x35, 0x06, 0xF1, 0x64, 0x5D, 0xA8, 0x62, 0x7C, 0xBA, 0x60, 0x32, 0x46,
100  0x8D, 0xCE, 0x61, 0xE7, 0xF0, 0x18, 0xFE, 0xA3, 0x68, 0xD0, 0xCB, 0x9D, 0x4B, 0x98, 0xB7, 0xC0,
101  0xF3, 0x72, 0xC9, 0xA7, 0x7A, 0x9F, 0xA4, 0x28, 0x8D, 0x55, 0x62, 0x51, 0x3F, 0x92, 0xB2, 0x1B,
102  0x26, 0xDD, 0x76, 0x45, 0x61, 0x29, 0x70, 0xE9, 0xF5, 0x1D, 0x51, 0xE1, 0x2D, 0xEB, 0x42, 0x3C,
103  0xF3, 0xFF, 0x69, 0xF1, 0xAA, 0xFA, 0x7F, 0xEF, 0xED, 0x92, 0x29, 0xBE, 0x6B, 0x4F, 0x5F, 0xDB,
104  0x2E, 0xE1, 0xBE, 0x7D, 0xA3, 0x53, 0x8C, 0x7A, 0x4B, 0xF4, 0x33, 0x20, 0xA1, 0x07, 0x8B, 0xAA,
105  0x00, 0x56, 0x57, 0x85, 0x3D, 0x8F, 0x34, 0x66, 0x03, 0x1B, 0xD8, 0x8E, 0xEB, 0xB6, 0x5C, 0x7F,
106  0x59, 0xDD, 0x89, 0x94, 0x7B, 0xF2, 0x8A, 0x9F, 0x56, 0xBB, 0xFC, 0xA6, 0xD2, 0x46, 0x90, 0xB9,
107  0xFE, 0x27, 0xA4, 0xFD, 0xBE, 0x03, 0xAE, 0x34, 0x34, 0x36, 0x2B, 0x14, 0x1B, 0xF8, 0xE3, 0x76,
108  0x13, 0xD5, 0x72, 0x25, 0xDE, 0x09, 0x96, 0x2E, 0xE6, 0xC2, 0xA5, 0x43, 0xCC, 0xE9, 0x06, 0x8B,
109  0xAA, 0xB9, 0x1F, 0x3A, 0xFC, 0x1A, 0x4D, 0x80, 0x5D, 0x44, 0x3B, 0xF8, 0x8D, 0x96, 0xDC, 0x95,
110  0x0B, 0x13, 0x99, 0xBC, 0x8E, 0x26, 0xCC, 0x74, 0xDF, 0xCE, 0x76, 0x49, 0x9B, 0xF7, 0x0E, 0xC5,
111  0xB9, 0x5A, 0xED, 0xD6, 0x15, 0xED, 0x7E, 0x36, 0x32, 0xD4, 0x83, 0xD6, 0xC8, 0xE8, 0x71, 0xD2,
112  0x9E, 0x04, 0xF8, 0xAA, 0x86, 0x32, 0x87, 0xE2, 0xDF, 0x01, 0xDF, 0x10, 0xDD, 0x7C, 0xED, 0xCF,
113  0x0A, 0x80, 0xF0, 0xA2, 0x76, 0x3F, 0xAC, 0x31, 0x75, 0xF1, 0x94, 0x75, 0xFC, 0x31, 0xD9, 0xEE,
114  0xD3, 0xF5, 0x1A, 0xE9, 0xA6, 0xA4, 0x84, 0xAA, 0x5D, 0x35, 0xAE, 0xBE, 0xCA, 0x93, 0x2E, 0xB6,
115  0x49, 0xC1, 0xD5, 0x0C, 0xD6, 0xA7, 0x1E, 0x97, 0x67, 0xB7, 0x96, 0xB4, 0x4F, 0x50, 0x99, 0x39,
116  0x92, 0x5D, 0xB9, 0x62, 0xEF, 0x53, 0x72, 0xA6, 0xD9, 0x73, 0x6D, 0xE6, 0xC5, 0x90, 0xC4, 0xED,
117  0x33, 0x83, 0x19, 0xC6, 0xCC, 0x44, 0xC9, 0x07, 0x78, 0xF0, 0xEF, 0x8D, 0xDE, 0x89, 0xF2, 0x62,
118  0x41, 0xC9, 0xE0, 0x32, 0xD1, 0x0A, 0xB7, 0xB5, 0x56, 0x4B, 0x56, 0xE1, 0x02, 0x59, 0xCE, 0x45,
119  0x4A, 0xA0, 0xCA, 0xA3, 0xB2, 0x2C, 0xC3, 0xAC, 0xEE, 0xD1, 0x94, 0x3A, 0xBB, 0x33, 0x43, 0xD0,
120  0xF6, 0x52, 0x77, 0x45, 0xC0, 0x51, 0x85, 0x5F, 0x1D, 0x77, 0x97, 0x36, 0x0D, 0xBB, 0x27, 0x9D,
121  0x8A, 0x29, 0x4D, 0xD7, 0x0F, 0xCB, 0xE3, 0x0A, 0x8D, 0xC1, 0x7E, 0x11, 0x51, 0x4C, 0x24, 0xB6,
122  0x2E, 0x95, 0x3B, 0x4F, 0xCC, 0xB9, 0x95, 0x67, 0x6A, 0x95, 0xEE, 0xD2, 0x98, 0x2A, 0x0F, 0xC7,
123  0x9C, 0xCD, 0xBB, 0x1D, 0x65, 0xA6, 0x2D, 0xDD, 0x7F, 0x23, 0xF8, 0x12, 0xD6, 0x9B, 0xF4, 0x82,
124  0xDC, 0x40, 0x33, 0xB1, 0x3A, 0x8D, 0x66, 0x8A, 0x5C, 0xB7, 0xCE, 0x26, 0x60, 0x29, 0x0E, 0x53,
125  0xC7, 0xB2, 0x70, 0xE9, 0x2E, 0x27, 0x5D, 0xF3, 0xE6, 0x58, 0xC4, 0x47, 0xC3, 0x23, 0xE9, 0x06,
126  0x96, 0xE1, 0x51, 0x7A, 0xAF, 0xCD, 0x90, 0xDE, 0xC5, 0x22, 0xDF, 0xE7, 0xC2, 0x2F, 0xB2, 0x68,
```

```
127  0xA6, 0x8B, 0xC2, 0x70, 0xA4, 0xD3, 0xBB, 0x31, 0xF4, 0xEC, 0x6D, 0x2F, 0x43, 0xCB, 0x59, 0x68,
128  0x8E, 0x35, 0xD2, 0x5D, 0x7F, 0xE2, 0xE7, 0xCE, 0xB1, 0xF3, 0x7C, 0xDB, 0x1B, 0x22, 0x75, 0xA4,
129  0x67, 0x2E, 0x09, 0xE8, 0x8C, 0x2A, 0xFD, 0x49, 0x37, 0xEE, 0xDD, 0x79, 0xF8, 0xE0, 0xC1, 0xC9,
130  0x4F, 0xF7, 0xBC, 0x71, 0x38, 0x17, 0xFF, 0xFD, 0x95, 0x5F, 0x41, 0x79, 0xF3, 0xBE, 0x77, 0xD2,
131  0x87, 0x86, 0x16, 0x13, 0xE2, 0x78, 0x93, 0x70, 0x78, 0xC4, 0x98, 0xE6, 0x04, 0x39, 0x02, 0x49,
132  0x0A, 0x64, 0x13, 0x09, 0x5D, 0x25, 0x5E, 0x3C, 0x24, 0x84, 0x1C, 0x35, 0x46, 0x81, 0x62, 0x08,
133  0x1B, 0xC6, 0xDB, 0x05, 0xD6, 0x69, 0xE9, 0x2C, 0xB1, 0x8D, 0xFD, 0xCB, 0xBC, 0x06, 0x4C, 0x29,
134  0x91, 0xF5, 0xC4, 0x28, 0x6C, 0x15, 0x31, 0x04, 0x32, 0x46, 0x4E, 0xAF, 0x87, 0x14, 0x8C, 0x49,
135  0xE4, 0x13, 0xD6, 0x97, 0xB6, 0xE7, 0xF9, 0xD4, 0x76, 0x80, 0x66, 0x98, 0x26, 0x22, 0xF1, 0x63,
136  0x31, 0x9B, 0x3C, 0x12, 0x09, 0xA5, 0x6E, 0xBC, 0xC3, 0x2C, 0x5C, 0x01, 0x65, 0xA5, 0x59, 0x57,
137  0xB8, 0x88, 0x0C, 0x9A, 0x99, 0x5F, 0x8F, 0x45, 0x14, 0x3B, 0xA6, 0x2D, 0xC4, 0xDC, 0xA6, 0x42,
138  0x20, 0xC6, 0xCE, 0x9F, 0x33, 0x07, 0x5B, 0x20, 0x37, 0x02, 0xD3, 0x76, 0x3B, 0xBA, 0xF1, 0xDB,
139  0xEF, 0xCF, 0x1F, 0x37, 0x20, 0x11, 0x75, 0x2E, 0xBB, 0xBD, 0x4E, 0xA7, 0x39, 0x3C, 0xE2, 0x43,
140  0xD6, 0xE6, 0xF5, 0x50, 0x37, 0x2E, 0x18, 0xAB, 0xDE, 0x29, 0xB0, 0xEA, 0xF4, 0xFA, 0x9B, 0xB3,
141  0x3A, 0xD5, 0x0D, 0xC6, 0x09, 0x98, 0x5C, 0x3E, 0x38, 0x39, 0xDD, 0x9C, 0xD1, 0x03, 0x90, 0xE9,
142  0x3D, 0x70, 0x3A, 0x05, 0xED, 0x4E, 0xB6, 0x51, 0xEE, 0x44, 0x37, 0x28, 0x1F, 0x88, 0x8A, 0xCB,
143  0xFE, 0xE9, 0x16, 0x7C, 0xEE, 0xEB, 0xA2, 0x24, 0x52, 0x97, 0x8D, 0x8F, 0x74, 0xE3, 0xEC, 0xE7,
144  0x67, 0x8D, 0x3E, 0xC8, 0xD8, 0x7B, 0x78, 0xB2, 0x39, 0xEF, 0xBE, 0x6E, 0xFC, 0x42, 0x85, 0x3C,
145  0xEE, 0x01, 0xA3, 0xFE, 0x16, 0x42, 0x1E, 0xEB, 0xC6, 0x0B, 0xC6, 0x09, 0xB8, 0x5C, 0x76, 0x1F,
146  0x6C, 0x21, 0x12, 0xB8, 0xD7, 0x2F, 0x8C, 0x13, 0xF8, 0x17, 0x75, 0xAF, 0x9A, 0x9C, 0x20, 0x5F,
147  0x32, 0xD3, 0x94, 0xC4, 0xE9, 0x6A, 0xF6, 0xC9, 0x9C, 0x2E, 0x0B, 0xE3, 0xBF, 0x23, 0x28, 0x1D,
148  0xE4, 0x6A, 0xED, 0x20, 0x16, 0x74, 0xA0, 0x12, 0x3F, 0xA8, 0x17, 0xBF, 0x92, 0x24, 0xC9, 0x65,
149  0x39, 0xDD, 0xE8, 0x76, 0x2A, 0x34, 0x60, 0xB4, 0x72, 0x16, 0x64, 0xC4, 0x19, 0x05, 0x74, 0xDA,
150  0x49, 0xB0, 0x18, 0xA6, 0xB7, 0x7E, 0x80, 0x8F, 0x1E, 0xEB, 0x52, 0x5C, 0x6F, 0x94, 0x22, 0x14,
151  0xD2, 0xA2, 0x4B, 0xDD, 0x38, 0x39, 0xAE, 0xB2, 0xF7, 0x16, 0x70, 0x8C, 0x59, 0x9B, 0xE2, 0xE1,
152  0x30, 0x5C, 0x1B, 0x91, 0x94, 0x54, 0x37, 0x9E, 0x24, 0xC7, 0xDB, 0xE0, 0xD2, 0xEA, 0x6D, 0x81,
153  0x8B, 0x24, 0x0E, 0x87, 0xA6, 0xD5, 0x13, 0xD0, 0xF4, 0xF4, 0x34, 0x22, 0xBE, 0x26, 0x30, 0x55,
154  0xD2, 0x6E, 0x83, 0x0B, 0x2D, 0xE2, 0x01, 0x0A, 0xC9, 0xDA, 0xA8, 0xC4, 0x84, 0x90, 0xD6, 0xC4,
155  0xD1, 0xDE, 0x10, 0x49, 0x44, 0xF9, 0x0E, 0xF0, 0x08, 0x11, 0x89, 0x02, 0x76, 0x43, 0xDC, 0xDA,
156  0x88, 0xA4, 0xA4, 0x50, 0x0F, 0x93, 0xE3, 0xBD, 0xA1, 0x22, 0x89, 0xF3, 0x3D, 0xE0, 0x32, 0xC7,
157  0xA6, 0x83, 0xDC, 0x8F, 0xD8, 0xB6, 0xA1, 0x64, 0xAD, 0x8F, 0x4D, 0x86, 0x1C, 0xF0, 0xE1, 0xDF,
158  0xB5, 0x73, 0xF6, 0x7D, 0xED, 0x1E, 0x31, 0xC7, 0xEE, 0x6B, 0x35, 0x8A, 0x1D, 0x75, 0xDF, 0xF2,
159  0xDA, 0x4F, 0xE4, 0xDC, 0xB0, 0x43, 0xE8, 0x02, 0x13, 0x3C, 0x61, 0x2B, 0xE5, 0x8D, 0x79, 0xF4,
160  0x74, 0xE3, 0x79, 0x80, 0xAE, 0xD8, 0xB3, 0x05, 0xDB, 0x34, 0x3D, 0xEF, 0xB0, 0xA5, 0xFD, 0x0A,
161  0x4B, 0xC1, 0x6D, 0x3A, 0xB0, 0xE7, 0x01, 0x86, 0x65, 0xE2, 0x56, 0x5C, 0xEE, 0x43, 0x31, 0x83,
162  0x83, 0xED, 0x98, 0x40, 0xC3, 0x7A, 0x81, 0xE7, 0x0E, 0xFA, 0x16, 0x1A, 0x2E, 0xB4, 0x1C, 0xAF,
163  0x1D, 0x16, 0x40, 0xA3, 0x1B, 0x8F, 0x3F, 0x3C, 0x59, 0x3B, 0x49, 0xF1, 0xFD, 0xE6, 0x3A, 0x1E,
164  0xCE, 0xB3, 0x93, 0x10, 0x50, 0x5F, 0x59, 0x6C, 0xAA, 0x23, 0xA7, 0xEE, 0x82, 0x53, 0xA1, 0x57,
165  0x2C, 0x20, 0xDB, 0x9E, 0xD3, 0x25, 0x35, 0xEB, 0xE9, 0x78, 0x73, 0x19, 0x0C, 0x84, 0xF8, 0x38,
166  0x41, 0xCE, 0xFA, 0x75, 0x25, 0x26, 0x64, 0x48, 0x69, 0xCF, 0xE1, 0x68, 0x57, 0x70, 0xF1, 0x69,
167  0xF7, 0x86, 0x99, 0xD0, 0x7A, 0xDF, 0xC0, 0x81, 0x20, 0x33, 0xDF, 0x5A, 0x7F, 0x3B, 0x42, 0xD0,
168  0xE9, 0x06, 0xA0, 0xF6, 0x0A, 0x0E, 0xD6, 0xAE, 0x32, 0x31, 0x83, 0x1B, 0x2E, 0x2F, 0x8F, 0x23,
169  0xE2, 0x6F, 0x53, 0x59, 0x2E, 0x22, 0xCF, 0xBB, 0xDA, 0xA6, 0xAC, 0x9C, 0xB9, 0x7E, 0x64, 0x6D,
170  0xCE, 0x01, 0x6A, 0xCA, 0x1B, 0xDB, 0x76, 0xCC, 0xCD, 0xAB, 0x12, 0x54, 0x94, 0x17, 0xFE, 0xAC,
171  0x26, 0xFD, 0x0D, 0x67, 0x71, 0x6C, 0xAE, 0x9F, 0x20, 0xB0, 0x09, 0x28, 0x9E, 0x9F, 0x69, 0x17,
172  0xE7, 0xAF, 0x2F, 0xDE, 0xBC, 0xDB, 0x4D, 0x76, 0x80, 0x39, 0xF7, 0x94, 0x18, 0xA8, 0xB6, 0xFB,
173  0xCE, 0x09, 0x20, 0x44, 0x6F, 0x13, 0x9C, 0x7A, 0x1C, 0xA8, 0xA7, 0x17, 0x6F, 0x77, 0x85, 0x52,
174  0x6F, 0x7F, 0x30, 0xF5, 0xBE, 0x05, 0x9C, 0x3E, 0xBA, 0x78, 0x81, 0xDD, 0x0D, 0xB0, 0xE2, 0x84,
175  0x14, 0x2F, 0xED, 0x25, 0x3D, 0xDA, 0xDB, 0x42, 0x2E, 0x11, 0xE5, 0x3B, 0x58, 0xC6, 0x81, 0x57,
176  0x7C, 0x64, 0x42, 0x6F, 0x12, 0x3C, 0x9C, 0x52, 0x37, 0xCE, 0x2F, 0xE7, 0x7E, 0x18, 0x05, 0x35,
177  0x0B, 0xAA, 0x1A, 0x91, 0x6D, 0x76, 0x06, 0x53, 0x51, 0x38, 0x22, 0xF1, 0xD6, 0x20, 0xDD, 0xD9,
178  0x4F, 0x30, 0xE9, 0x75, 0xFA, 0x5F, 0x15, 0x15, 0xCA, 0xFC, 0x26, 0x81, 0x99, 0x6C, 0x50, 0x77,
179  0x26, 0xB4, 0xEE, 0x3C, 0x3F, 0xDB, 0x4D, 0x2A, 0x9B, 0xEC, 0xAD, 0xE0, 0x4C, 0xF6, 0x5A, 0x70,
180  0x34, 0x7E, 0x51, 0x34, 0x81, 0x69, 0xC3, 0x45, 0x84, 0x20, 0x84, 0xB5, 0xF3, 0x26, 0x0B, 0x08,
181  0x79, 0x53, 0xFD, 0x72, 0x9B, 0xD0, 0x89, 0xC5, 0xC8, 0x46, 0xCE, 0x71, 0x1A, 0x37, 0xF7, 0xBF,
182  0x6A, 0xD4, 0x1C, 0x57, 0x4A, 0xBB, 0x4D, 0xD0, 0x50, 0x4D, 0x4C, 0xEC, 0xB8, 0xF4, 0x09, 0xA6,
183  0x75, 0x01, 0x91, 0x68, 0x39, 0x26, 0xDA, 0x19, 0xFF, 0xB6, 0x0D, 0x36, 0xBD, 0x6D, 0xB0, 0x91,
184  0x25, 0xCA, 0xC2, 0x73, 0x72, 0x43, 0x95, 0xA6, 0xDB, 0x3B, 0xBD, 0x49, 0x78, 0xC6, 0xF3, 0xF5,
185  0x73, 0x1A, 0xD0, 0xE8, 0xC6, 0x93, 0xB7, 0xBB, 0xC9, 0x69, 0x74, 0xB2, 0x9A, 0x39, 0x6D, 0xAB,
186  0x0C, 0xC6, 0x94, 0xDA, 0x77, 0x2B, 0xB6, 0xDC, 0x00, 0x8D, 0x25, 0x15, 0xFC, 0xC3, 0x8E, 0xD0,
187  0x58, 0xD6, 0x47, 0xE3, 0x2B, 0x57, 0x98, 0xE5, 0xB7, 0x80, 0x4F, 0x80, 0x96, 0x1F, 0x27, 0x33,
188  0xB4, 0x36, 0x46, 0x82, 0x4E, 0x37, 0xDE, 0xA1, 0xA5, 0xF6, 0xFC, 0xD5, 0xE3, 0x9D, 0x60, 0x15,
189  0x4F, 0xBA, 0x1F, 0xBC, 0x12, 0x95, 0xF7, 0x8D, 0x99, 0x8B, 0xBD, 0xF5, 0x83, 0x8A, 0x12, 0xE9,
190  0xC6, 0x4B, 0xEC, 0x85, 0xDA, 0x99, 0x1F, 0x88, 0xB7, 0xCD, 0xEC, 0x04, 0x35, 0x36, 0xF3, 0x7E,
191  0x20, 0xE3, 0x4A, 0xEF, 0x1B, 0xAF, 0xE9, 0xCC, 0x09, 0x02, 0x3F, 0x58, 0x1B, 0x32, 0x41, 0xA7,
192  0x1B, 0x2F, 0x5A, 0xAF, 0xD8, 0xD1, 0x4E, 0xE0, 0x8A, 0x67, 0xDD, 0x0F, 0x62, 0x89, 0xCE, 0xFB,
193  0x06, 0x6D, 0x61, 0xBB, 0xCE, 0x7C, 0x6D, 0xC8, 0x18, 0x95, 0x6E, 0xBC, 0x6F, 0x3D, 0x83, 0x7F,
194  0x77, 0x02, 0x17, 0x9F, 0x71, 0x3F, 0x60, 0x09, 0x6D, 0xF7, 0x0D, 0x95, 0x65, 0x2E, 0xD7, 0x06,
195  0x0A, 0x68, 0x74, 0xE3, 0xE9, 0xD9, 0x07, 0xAD, 0xF1, 0xD4, 0x5F, 0x7A, 0xF4, 0xC6, 0x3F, 0xED,
196  0xFC, 0x75, 0x73, 0x27, 0x88, 0xD1, 0xA9, 0xF7, 0x83, 0x17, 0x53, 0x7A, 0xDF, 0x68, 0xB1, 0xBB,
197  0x8F, 0xC7, 0x68, 0xFD, 0x74, 0x18, 0x13, 0xD2, 0x7B, 0x5F, 0xE0, 0x48, 0x7B, 0x82, 0x76, 0x93,
198  0x10, 0x93, 0x79, 0x77, 0xD1, 0xB4, 0xA7, 0x4A, 0xEE, 0x1B, 0x27, 0x1B, 0x99, 0xF8, 0xA3, 0x85,
199  0xC9, 0x26, 0x37, 0x5E, 0x48, 0xB4, 0xBA, 0xF1, 0x0C, 0xBE, 0x68, 0x4F, 0xD9, 0x97, 0x5D, 0xB5,
200  0x1C, 0xF2, 0xFC, 0xBB, 0x40, 0x2D, 0xA3, 0xEF, 0x37, 0x01, 0x1C, 0x34, 0x78, 0xFE, 0xC4, 0xDB,
201  0xE8, 0x7E, 0xEA, 0x0C, 0xB9, 0x80, 0xEF, 0x1D, 0xFF, 0xBE, 0x5B, 0x00, 0x53, 0x21, 0x76, 0x86,
202  0xA1, 0xA4, 0xF7, 0x2E, 0x60, 0x8C, 0x9F, 0x49, 0x60, 0xDB, 0x02, 0xFC, 0xE5, 0x4F, 0x55, 0x48,
203  0x89, 0x57, 0xC2, 0xB0, 0xAD, 0x1B, 0x4C, 0x5A, 0x21, 0x71, 0x5C, 0x57, 0x37, 0x9E, 0x63, 0xA2,
204  0x5D, 0xD0, 0xC3, 0xE1, 0x11, 0x1F, 0x50, 0x9F, 0x8B, 0xB8, 0xE1, 0x9F, 0xBE, 0x76, 0x0D, 0xCD,
205  0x74, 0xE3, 0x82, 0xBE, 0x16, 0x0B, 0x78, 0xD1, 0x6F, 0xEB, 0x33, 0x63, 0x46, 0xC4, 0x5E, 0xE0,
206  0x83, 0x50, 0x09, 0x48, 0xE2, 0xED, 0x24, 0xBA, 0x16, 0x1F, 0x49, 0xBF, 0x19, 0xE7, 0x6C, 0xB0,
207  0x46, 0xBD, 0xAC, 0x7A, 0x3A, 0x7A, 0x15, 0xD6, 0x2C, 0xBE, 0x58, 0x3B, 0x3C, 0xF2, 0x90, 0xC2,
208  0xDC, 0x05, 0x28, 0x0C, 0xF9, 0xFB, 0xD4, 0x0A, 0x58, 0x25, 0x0F, 0x53, 0x30, 0x4B, 0xA4, 0x0F,
209  0x26, 0x25, 0x6A, 0xE5, 0x1F, 0x58, 0x12, 0x1B, 0xB6, 0xF5, 0x82, 0x96, 0x3D, 0x7A, 0x24, 0xEA,
210  0x21, 0x3D, 0x4C, 0xCC, 0xFF, 0x9F, 0x7F, 0x57, 0xF9, 0x0C, 0x7D, 0xDB, 0x5D, 0x2A, 0x98, 0xAE,
211  0x85, 0x81, 0x39, 0xD2, 0x8B, 0x1E, 0xCD, 0x28, 0xD0, 0xFC, 0x48, 0xA5, 0x7A, 0x6E, 0xB0, 0xC2,
212  0xD6, 0xC3, 0xD0, 0x0C, 0x9C, 0x39, 0x31, 0x6E, 0x59, 0xBE, 0x19, 0xCD, 0xB0, 0x47, 0xDA, 0xC8,
213  0xB2, 0xCE, 0x17, 0x70, 0xF0, 0xD2, 0x09, 0x09, 0x06, 0x2B, 0x34, 0x0E, 0x9E, 0xBE, 0x79, 0x75,
```

```
214  0xC6, 0x1F, 0x51, 0x79, 0xE9, 0x23, 0x0B, 0x5B, 0x07, 0x87, 0x9A, 0x1D, 0x79, 0xDC, 0xCD, 0x1B,
215  0x98, 0x8E, 0xE5, 0x6F, 0x1A, 0x5C, 0xA0, 0x40, 0x1B, 0xA3, 0x10, 0xBF, 0xF0, 0x43, 0xA2, 0x8D,
216  0xB4, 0x84, 0xA3, 0xEB, 0x9B, 0xEC, 0xF6, 0xC5, 0xB6, 0x1F, 0x38, 0x13, 0xC7, 0x13, 0x23, 0xB9,
217  0xB2, 0xBF, 0x05, 0x2E, 0x0C, 0x4D, 0xA8, 0x7E, 0xD4, 0x0E, 0x06, 0xA7, 0xDD, 0x03, 0xFA, 0x34,
218  0x11, 0xC0, 0x00, 0x3F, 0x00, 0x04, 0x18, 0x06, 0x40, 0x80, 0x8F, 0x0C, 0xF1, 0x38, 0x11, 0x76,
219  0xDB, 0xCC, 0xE4, 0x54, 0x40, 0x2A, 0x6D, 0xE3, 0x80, 0xE3, 0x74, 0x40, 0x1F, 0xAD, 0xBB, 0x4E,
220  0x28, 0xC3, 0xA9, 0xBF, 0x2C, 0xA3, 0x0C, 0xF0, 0xCC, 0x5F, 0xE0, 0x1C, 0x71, 0x42, 0x2D, 0xBC,
221  0xB9, 0x72, 0xEA, 0xD8, 0xEB, 0x0F, 0x9A, 0xF1, 0x80, 0xE4, 0xCD, 0x3D, 0x23, 0x8D, 0x04, 0x11,
222  0xCE, 0xB2, 0xC5, 0x5E, 0x15, 0xD7, 0x58, 0xAC, 0x52, 0xC6, 0x36, 0x72, 0xC3, 0x1C, 0xE7, 0x68,
223  0x6E, 0x21, 0x82, 0xDF, 0xD3, 0xDD, 0x5D, 0x18, 0xD0, 0xC0, 0xEE, 0x21, 0xDF, 0xEA, 0x3D, 0x14,
224  0x67, 0xDE, 0x01, 0x5F, 0x82, 0x9B, 0xE9, 0xAC, 0xF2, 0xCF, 0x40, 0x91, 0xFD, 0x3A, 0xD2, 0xBC,
225  0x08, 0x42, 0xF8, 0x11, 0x53, 0x41, 0x1B, 0x64, 0xCE, 0x32, 0x6A, 0x17, 0xB2, 0x93, 0x78, 0x4B,
226  0x31, 0x9B, 0x93, 0xFD, 0xE8, 0xD8, 0x74, 0xE2, 0x36, 0x7B, 0x67, 0xF2, 0x08, 0x78, 0x1C, 0xC4,
227  0xD9, 0xFD, 0x20, 0x7D, 0x15, 0xA5, 0x4C, 0xC4, 0xEC, 0xD0, 0x16, 0x7D, 0xB0, 0x38, 0xBF, 0x10,
228  0x27, 0x6E, 0xDF, 0x5E, 0x24, 0x7C, 0x35, 0x69, 0x18, 0x9C, 0x4A, 0x4F, 0x5C, 0xC3, 0x09, 0xE9,
229  0x79, 0xBF, 0x55, 0xDE, 0x39, 0x1E, 0x31, 0x73, 0x89, 0xC3, 0xAD, 0x44, 0xF2, 0x8C, 0x05, 0xEE,
230  0xDD, 0xCB, 0x72, 0xBB, 0x3D, 0x12, 0x54, 0xA9, 0x26, 0x7C, 0x3C, 0x44, 0x06, 0x44, 0x1E, 0xA8,
231  0x2D, 0x9E, 0x02, 0x15, 0x22, 0x39, 0x76, 0xE3, 0x76, 0xC6, 0xF0, 0x89, 0x8C, 0x36, 0x35, 0x91,
232  0x63, 0x31, 0x03, 0xB1, 0x7B, 0x20, 0x9A, 0xE9, 0x53, 0x72, 0x5C, 0xBE, 0x47, 0xCC, 0xEB, 0x1B,
233  0x58, 0x5C, 0x1D, 0x6D, 0x82, 0xFD, 0xA9, 0x33, 0xA7, 0x3F, 0x88, 0xF1, 0xE9, 0x54, 0x32, 0xC7,
234  0x49, 0x86, 0x23, 0x55, 0x2C, 0x27, 0x37, 0xFD, 0x30, 0x7E, 0xF4, 0x3A, 0x81, 0xB8, 0x56, 0x21,
235  0x3F, 0x95, 0xCA, 0x26, 0x07, 0x36, 0xF4, 0x5A, 0x46, 0xFA, 0x7B, 0xCE, 0xD4, 0xC9, 0xC0, 0x02,
236  0x26, 0x6C, 0x82, 0x55, 0x26, 0xA5, 0x92, 0xC7, 0x37, 0x8A, 0x29, 0x0C, 0xC2, 0xD8, 0x2D, 0xC7,
237  0xD4, 0x14, 0x6C, 0x56, 0x38, 0x2C, 0x63, 0x95, 0x2B, 0xFC, 0x0A, 0x86, 0x3C, 0x10, 0x1B, 0xBC,
238  0xAE, 0x3D, 0x61, 0x35, 0x8A, 0x32, 0x17, 0x31, 0x96, 0xFD, 0xFD, 0x96, 0x2C, 0xFC, 0x75, 0x1C,
239  0x76, 0x49, 0x0A, 0x94, 0xFD, 0x80, 0xFA, 0x7F, 0x6C, 0x69, 0x1A, 0x22, 0xA9, 0xA3, 0x89, 0x07,
240  0xFB, 0xE3, 0xF8, 0x48, 0xE1, 0x30, 0x21, 0xF7, 0x49, 0x91, 0x32, 0xC8, 0x89, 0x2A, 0x87, 0x08,
241  0xC8, 0xDD, 0xD5, 0xE4, 0x47, 0xF5, 0xC7, 0x90, 0x42, 0x3F, 0x67, 0xF8, 0xB0, 0x8B, 0x32, 0x09,
242  0x13, 0xFE, 0x1B, 0xBF, 0xCD, 0xA9, 0xE5, 0x7B, 0x58, 0xCD, 0x5D, 0x0E, 0x12, 0x15, 0x4F, 0x5E,
243  0xC2, 0xF3, 0x4C, 0xA3, 0xF1, 0xCC, 0x21, 0x0A, 0x86, 0x07, 0x90, 0xBE, 0x55, 0xBC, 0x44, 0x63,
244  0x97, 0x12, 0x04, 0x98, 0x44, 0x81, 0x27, 0x47, 0x21, 0xCF, 0x64, 0x7F, 0x47, 0x38, 0xB8, 0x02,
245  0x46, 0x9F, 0xEE, 0x7E, 0x89, 0xEB, 0xC2, 0xF5, 0x11, 0x7B, 0x34, 0xC1, 0x77, 0x1F, 0x41, 0xE5,
246  0x18, 0xDD, 0xFD, 0xC2, 0xA0, 0xBE, 0xBE, 0x07, 0x53, 0xC2, 0x17, 0x36, 0xF1, 0xF5, 0x27, 0xCE,
247  0xC2, 0xA6, 0x2F, 0x9A, 0x6D, 0x30, 0x16, 0x31, 0x6E, 0x6D, 0x32, 0xC5, 0x5E, 0x23, 0xC0, 0xE1,
248  0x1C, 0xD8, 0xE3, 0x34, 0x01, 0xC6, 0x33, 0xFA, 0x2E, 0x86, 0x12, 0x35, 0x69, 0x7C, 0x0A, 0x30,
249  0xD0, 0x81, 0x00, 0xC4, 0xD7, 0xEE, 0x7E, 0x61, 0x2C, 0xAE, 0x35, 0x1B, 0xB2, 0x40, 0x38, 0xC5,
250  0xD6, 0x21, 0xD4, 0x2B, 0x44, 0xE8, 0x13, 0xB8, 0x77, 0xBF, 0xC4, 0xAC, 0xDA, 0xFC, 0xA7, 0xEB,
251  0x4F, 0x89, 0x87, 0x24, 0x45, 0x24, 0xAE, 0x7D, 0xEC, 0x44, 0x9B, 0xF1, 0xBA, 0x60, 0x28, 0xF8,
252  0xC1, 0x63, 0xD7, 0x6D, 0x1C, 0xF0, 0x07, 0x95, 0x45, 0x6E, 0x6F, 0x43, 0xB3, 0x7A, 0x8E, 0x40,
253  0x6C, 0xB9, 0x28, 0xB0, 0x7C, 0xE5, 0x7B, 0xA6, 0xEB, 0x98, 0x9F, 0x69, 0x42, 0x6F, 0x66, 0x05,
254  0xE7, 0x19, 0xC2, 0x6D, 0xF3, 0x17, 0xCF, 0xBC, 0xF6, 0x2D, 0x9C, 0x73, 0xD3, 0x26, 0x15, 0xE3,
255  0xE8, 0x08, 0xAC, 0x8C, 0xAC, 0x38, 0x95, 0x71, 0x8C, 0xE8, 0x1B, 0x0A, 0xB8, 0x99, 0x32, 0x16,
256  0xE6, 0xCA, 0x08, 0x5D, 0xB8, 0xCD, 0xD2, 0x2A, 0x1F, 0xAB, 0x9C, 0xBA, 0x2D, 0x47, 0x4F, 0x4B,
257  0x6C, 0xF1, 0x57, 0xE8, 0x7B, 0x8D, 0xE6, 0xAD, 0xC4, 0x0C, 0xAB, 0x3C, 0xE8, 0x04, 0x12, 0x83,
258  0x8C, 0x89, 0x8A, 0xCC, 0x94, 0x5D, 0x0D, 0x1C, 0xA4, 0x99, 0xA4, 0xC0, 0x66, 0xF4, 0x23, 0x55,
259  0x42, 0x56, 0x06, 0xD9, 0xBC, 0x7F, 0x30, 0x97, 0xF9, 0xF3, 0x90, 0x97, 0x4E, 0x29, 0x23, 0x35,
260  0x25, 0x73, 0x71, 0xFF, 0xA3, 0xAF, 0xE8, 0x97, 0xDB, 0x17, 0xE8, 0xC9, 0xCF, 0x5D, 0x4C, 0x0F,
261  0x9F, 0x5C, 0xFD, 0x0C, 0x25, 0x9F, 0x37, 0x2E, 0x4C, 0x96, 0x94, 0xE0, 0x2C, 0x69, 0x1A, 0x2B,
262  0x29, 0xD3, 0x06, 0x53, 0xE2, 0xC1, 0x9A, 0x7E, 0x9E, 0x6F, 0xCA, 0x38, 0x24, 0xEB, 0x83, 0x0C,
263  0x29, 0xE5, 0x5A, 0x4D, 0x9B, 0x59, 0x15, 0x48, 0xF4, 0x72, 0xAE, 0x2B, 0xA3, 0x97, 0x16, 0x02,
264  0x12, 0x35, 0x73, 0xE4, 0x6A, 0x62, 0xB9, 0x25, 0x3E, 0x90, 0x8C, 0x1D, 0x12, 0x7F, 0xCE, 0x57,
265  0x26, 0x39, 0x27, 0x5F, 0x3A, 0x9E, 0xE5, 0x2F, 0xDB, 0xF4, 0x7C, 0x43, 0x94, 0x56, 0x59, 0xD1,
266  0xB6, 0xE3, 0x81, 0x01, 0x5F, 0xFC, 0xFA, 0xEA, 0x25, 0x4D, 0x39, 0xF2, 0x0A, 0xE7, 0x20, 0xDB,
267  0x17, 0xB1, 0x77, 0x02, 0x2B, 0x67, 0xA0, 0xB0, 0xB5, 0xA1, 0xD5, 0xE6, 0xA9, 0x26, 0x69, 0x47,
268  0x69, 0x24, 0xD0, 0xC3, 0x4F, 0x7C, 0x4E, 0x5A, 0x78, 0x32, 0x00, 0x37, 0x2B, 0x65, 0xF1, 0xE7,
269  0x79, 0x51, 0x20, 0x0E, 0x1F, 0x13, 0x02, 0xEE, 0xAA, 0x71, 0x47, 0x0E, 0x69, 0x8E, 0x11, 0xAB,
270  0xC3, 0x5B, 0x9A, 0x0C, 0x7E, 0x41, 0xC8, 0xA7, 0x66, 0x12, 0x31, 0x96, 0x15, 0x5E, 0xCA, 0x93,
271  0x68, 0x0E, 0x71, 0x89, 0x1F, 0x7D, 0x34, 0xC7, 0x90, 0x1A, 0x9F, 0x82, 0xE7, 0xB7, 0x3D, 0xD0,
272  0xA0, 0x79, 0x5D, 0xA6, 0x0E, 0x37, 0x57, 0x0A, 0x64, 0x5D, 0x21, 0x58, 0x12, 0x52, 0x73, 0xCB,
273  0xD8, 0x47, 0xCD, 0x4E, 0xF6, 0xDE, 0x73, 0x2F, 0x6E, 0x6D, 0x8B, 0x0C, 0x3B, 0x5A, 0x35, 0x2D,
274  0xEF, 0x6E, 0x32, 0x0C, 0xD2, 0xF4, 0xB2, 0x22, 0x6C, 0xAE, 0x81, 0x91, 0xFC, 0x22, 0x1E, 0x10,
275  0xCB, 0x2E, 0x07, 0x44, 0x81, 0xEC, 0xD9, 0xDE, 0x2F, 0xD7, 0x2C, 0xE4, 0x20, 0x17, 0x39, 0x4C,
276  0xA3, 0x2F, 0x2A, 0x98, 0xD2, 0xF2, 0x2C, 0x9C, 0xA0, 0x4E, 0x99, 0x50, 0xE6, 0xBF, 0xD2, 0x7A,
277  0xC1, 0x67, 0x88, 0xA5, 0xCD, 0xF7, 0xA8, 0xD9, 0xDA, 0x70, 0x16, 0x81, 0x95, 0x66, 0xB1, 0x4F,
278  0xF2, 0xDF, 0x68, 0xC3, 0x96, 0x04, 0x0F, 0x34, 0x70, 0x65, 0x41, 0x0D, 0xA7, 0xA5, 0x4C, 0x20,
279  0xBA, 0xBD, 0x0A, 0x02, 0xE9, 0xAE, 0x27, 0x89, 0x56, 0xEA, 0x22, 0x4B, 0xD3, 0x5F, 0xFE, 0x3E,
280  0x1D, 0xC6, 0x02, 0xB8, 0xAE, 0x6A, 0xAE, 0xC0, 0x09, 0xC6, 0x35, 0x13, 0xB7, 0xA1, 0x44, 0xA2,
281  0xAD, 0x92, 0x9C, 0xA6, 0xA0, 0x2D, 0x5E, 0x6D, 0x89, 0x73, 0xDE, 0x54, 0xD4, 0x0A, 0xAF, 0xB6,
282  0xC1, 0xD7, 0x92, 0x83, 0xC4, 0xF7, 0x3F, 0xA6, 0x26, 0xC4, 0xE5, 0xF6, 0xC6, 0xB2, 0xBD, 0xE3,
283  0xE5, 0x40, 0x05, 0x85, 0x7C, 0x9B, 0x26, 0x37, 0x17, 0xAE, 0x69, 0x2E, 0x2C, 0xCC, 0x45, 0x09,
284  0xD2, 0x0E, 0xB4, 0x7A, 0x6D, 0x92, 0xF8, 0xFF, 0x87, 0x27, 0xA9, 0x66, 0xCB, 0x71, 0xA9, 0x9C,
285  0xA2, 0xF7, 0x97, 0xD4, 0x2B, 0x27, 0xC8, 0x3C, 0xCB, 0xC1, 0xD5, 0x5A, 0x8E, 0xEB, 0xA9, 0x15,
286  0xAF, 0x1D, 0x28, 0x41, 0xAA, 0x96, 0x7A, 0x85, 0x11, 0xAB, 0x92, 0xEC, 0x75, 0xB3, 0xFF, 0xDD,
287  0x42, 0xF2, 0x66, 0x89, 0x44, 0x58, 0xBE, 0x51, 0x5C, 0x59, 0x3D, 0xF9, 0x30, 0x49, 0xC9, 0x64,
288  0x8D, 0x52, 0x49, 0x9A, 0x8C, 0x94, 0xA8, 0x13, 0x39, 0x4A, 0xA9, 0xE3, 0x41, 0xBC, 0xEC, 0x26,
289  0x5F, 0x6B, 0x19, 0x2B, 0x19, 0x9D, 0x06, 0x4E, 0xCA, 0x80, 0x77, 0xFC, 0x86, 0x76, 0x3F, 0xBF,
290  0x26, 0xE6, 0xBD, 0x17, 0x57, 0x36, 0xD7, 0x71, 0xC9, 0x03, 0x12, 0x95, 0x32, 0x63, 0x92, 0x00,
291  0xE1, 0xF4, 0x45, 0x62, 0x56, 0x8A, 0x82, 0x5C, 0x1C, 0x90, 0x86, 0xFE, 0xD6, 0xC5, 0x74, 0xBD,
292  0x22, 0x9E, 0xC6, 0x39, 0xFB, 0xF9, 0x99, 0xE6, 0x07, 0x1A, 0x7F, 0xC1, 0x5D, 0x90, 0xBC, 0x5B,
293  0x44, 0x13, 0x6F, 0x7F, 0x62, 0xAB, 0x42, 0x9A, 0x83, 0xC8, 0xD4, 0x09, 0xA1, 0x49, 0xA6, 0x4F,
294  0xDE, 0xE2, 0xDB, 0x7A, 0xF2, 0x82, 0xA7, 0x4A, 0xF5, 0x78, 0x57, 0xFC, 0x53, 0xA2, 0x48, 0xCE,
295  0x9C, 0x9C, 0x26, 0xB5, 0xE5, 0x6D, 0xA1, 0xE3, 0x4A, 0x22, 0x2A, 0x5B, 0x87, 0xAE, 0x61, 0xC2,
296  0xE4, 0xF4, 0x37, 0x6B, 0x45, 0xB5, 0x02, 0x95, 0x86, 0x4C, 0xC8, 0x52, 0x5B, 0xA6, 0xBA, 0xAE,
297  0x58, 0x53, 0xB5, 0xD8, 0x2F, 0x41, 0x94, 0xEE, 0x79, 0x29, 0xB3, 0x7C, 0x31, 0x2A, 0xDC, 0xE2,
298  0xBC, 0xB0, 0xF2, 0xCF, 0xF0, 0x28, 0xDE, 0x59, 0xE5, 0xDF, 0xF8, 0xAB, 0x8B, 0x86, 0x47, 0xFC,
299  0x7F, 0x22, 0xF6, 0x5F, 0x04, 0x9C, 0x39, 0x76, 0x5C, 0x6C, 0x00, 0x00
300 };
```

```
#endif

//File: index_ov3660.html.gz, Size: 4408
#define index_ov3660_html_gz_len 4408
const uint8_t index_ov3660_html_gz[] = {
 0x1F, 0x8B, 0x08, 0x08, 0x28, 0x5C, 0xAE, 0x5C, 0x00, 0x03, 0x69, 0x6E, 0x64, 0x65, 0x78, 0x5F,
 0x6F, 0x76, 0x33, 0x36, 0x36, 0x30, 0x2E, 0x68, 0x74, 0x6D, 0x6C, 0x00, 0xE5, 0x5D, 0xEB, 0x92,
 0xD3, 0xC6, 0x12, 0xFE, 0xCF, 0x53, 0x08, 0x41, 0x58, 0x6F, 0x65, 0xED, 0xF5, 0x6D, 0xCD, 0xE2,
 0xD8, 0xE6, 0xC0, 0xB2, 0x84, 0x54, 0x01, 0x49, 0x20, 0x21, 0xA9, 0x4A, 0xA5, 0x60, 0x2C, 0x8D,
 0xED, 0x09, 0xB2, 0xE4, 0x48, 0x23, 0x7B, 0x37, 0xD4, 0x3E, 0xC7, 0x79, 0xA0, 0xF3, 0x62, 0xA7,
 0xE7, 0x22, 0x69, 0x24, 0x8F, 0x2E, 0xB6, 0x59, 0x9B, 0xC3, 0x31, 0x55, 0x20, 0x5B, 0xD3, 0x3D,
 0xDD, 0xFD, 0xF5, 0x6D, 0x46, 0x17, 0x06, 0x77, 0x6D, 0xCF, 0xA2, 0xD7, 0x0B, 0x6C, 0xCC, 0xE8,
 0xDC, 0x19, 0xDD, 0x19, 0x88, 0x7F, 0x0C, 0xF8, 0x0C, 0x66, 0x18, 0xD9, 0xE2, 0x90, 0x7F, 0x9D,
 0x63, 0x8A, 0x0C, 0x6B, 0x86, 0xFC, 0x00, 0xD3, 0xA1, 0x19, 0xD2, 0x49, 0xFD, 0xDC, 0xCC, 0x9E,
 0x76, 0xD1, 0x1C, 0x0F, 0xCD, 0x25, 0xC1, 0xAB, 0x85, 0xE7, 0x53, 0xD3, 0xB0, 0x3C, 0x97, 0x62,
 0x17, 0x86, 0xAF, 0x88, 0x4D, 0x67, 0x43, 0x1B, 0x2F, 0x89, 0x85, 0xEB, 0xFC, 0xCB, 0x09, 0x71,
 0x09, 0x25, 0xC8, 0xA9, 0x07, 0x16, 0x72, 0xF0, 0xB0, 0xA5, 0xF2, 0xA2, 0x84, 0x3A, 0x78, 0x74,
 0xF9, 0xF6, 0xA7, 0x4E, 0xDB, 0xF8, 0xF1, 0x5D, 0xA7, 0xD7, 0x6B, 0x0E, 0x4E, 0xC5, 0x6F, 0xC9,
 0x98, 0x80, 0x5E, 0xAB, 0xDF, 0xD9, 0x67, 0xEC, 0xD9, 0xD7, 0xC6, 0xA7, 0xD4, 0x4F, 0xEC, 0x33,
 0x01, 0x21, 0xEA, 0x13, 0x34, 0x27, 0xCE, 0x75, 0xDF, 0x78, 0xE2, 0xC3, 0x9C, 0x27, 0x2F, 0xB0,
 0xB3, 0xC4, 0x94, 0x58, 0xE8, 0x24, 0x40, 0x6E, 0x50, 0x0F, 0xB0, 0x4F, 0x26, 0xDF, 0xAD, 0x11,
 0x8E, 0x91, 0xF5, 0x71, 0xEA, 0x7B, 0xA1, 0x6B, 0xF7, 0x8D, 0x7B, 0xAD, 0x73, 0xF6, 0x67, 0x7D,
 0x90, 0xE5, 0x39, 0x9E, 0x0F, 0xE7, 0x2F, 0x9F, 0xB3, 0x3F, 0xEB, 0xE7, 0xF9, 0xEC, 0x01, 0xF9,
 0x07, 0xF7, 0x8D, 0x56, 0x6F, 0x71, 0x95, 0x3A, 0x7F, 0x73, 0x27, 0xF5, 0x75, 0xD6, 0xCE, 0x93,
 0x5E, 0xD2, 0x9F, 0x17, 0xD3, 0x07, 0xD8, 0xA2, 0xC4, 0x73, 0x1B, 0x73, 0x44, 0x5C, 0x0D, 0x27,
 0x9B, 0x04, 0x0B, 0x07, 0x81, 0x0D, 0x26, 0x0E, 0x2E, 0xE4, 0x73, 0x6F, 0x8E, 0xDD, 0xF0, 0xA4,
 0x84, 0x1B, 0x63, 0x52, 0xB7, 0x89, 0x2F, 0x46, 0xF5, 0x99, 0x1D, 0xC2, 0xB9, 0x5B, 0xCA, 0xB6,
 0x48, 0x2E, 0xD7, 0x73, 0xB1, 0xC6, 0x80, 0x6C, 0xA2, 0x95, 0x8F, 0x16, 0x6C, 0x00, 0xFB, 0x77,
 0x7D, 0xC8, 0x9C, 0xB8, 0xC2, 0xA9, 0xFA, 0x46, 0xA7, 0xDB, 0x5C, 0x5C, 0x95, 0x40, 0xD9, 0xE9,
 0xB1, 0x3F, 0xEB, 0x83, 0x16, 0xC8, 0xB6, 0x89, 0x3B, 0xED, 0x1B, 0xE7, 0x5A, 0x16, 0x9E, 0x6F,
 0x63, 0xBF, 0xEE, 0x23, 0x9B, 0x84, 0x41, 0xDF, 0xE8, 0xEA, 0xC6, 0xCC, 0x91, 0x3F, 0x05, 0x59,
 0xA8, 0x07, 0xC2, 0xD6, 0x5B, 0x5A, 0x49, 0xE4, 0x10, 0x9F, 0x4C, 0x67, 0x14, 0x20, 0x5D, 0x1B,
 0x93, 0x35, 0x9A, 0x0C, 0xA1, 0x32, 0x3C, 0x0B, 0xED, 0xA6, 0xB7, 0x1A, 0x72, 0xC8, 0xD4, 0xAD,
 0x13, 0x8A, 0xE7, 0xA0, 0x4E, 0x40, 0x7D, 0x4C, 0xAD, 0x59, 0x91, 0x28, 0x13, 0x32, 0x0D, 0x7D,
 0xAC, 0x11, 0x24, 0xB6, 0x5B, 0x81, 0xC2, 0x70, 0x72, 0xFD, 0x54, 0x7D, 0x85, 0xC7, 0x1F, 0x09,
 0xAD, 0x4B, 0x9B, 0x8C, 0xF1, 0xC4, 0xF3, 0xB1, 0x76, 0x64, 0x34, 0xC2, 0xF1, 0xAC, 0x8F, 0xF5,
 0x80, 0x22, 0x9F, 0x56, 0x61, 0x88, 0x26, 0x14, 0xFB, 0xE5, 0xFC, 0x30, 0xF3, 0x8A, 0x72, 0x6E,
 0xF9, 0xD3, 0xCA, 0x01, 0xC4, 0x75, 0x88, 0x8B, 0xAB, 0x8B, 0x97, 0x37, 0x6F, 0x9A, 0x9D, 0x18,
 0x55, 0x01, 0x18, 0x32, 0x9F, 0x16, 0x79, 0x09, 0xD7, 0x75, 0x7D, 0x32, 0x19, 0x37, 0xAD, 0x66,
 0xF3, 0x9B, 0xF5, 0x93, 0x33, 0x2C, 0xDC, 0x14, 0x85, 0xD4, 0xDB, 0x3D, 0x22, 0xD6, 0xC2, 0x2A,
 0xA3, 0xC7, 0xBF, 0xE6, 0xD8, 0x26, 0xC8, 0xA8, 0x29, 0xE1, 0x7C, 0xDE, 0x04, 0x9F, 0x3A, 0x36,
 0x90, 0x6B, 0x1B, 0x35, 0xCF, 0x27, 0x10, 0x08, 0x88, 0xA7, 0x1B, 0x07, 0x7E, 0x81, 0xC2, 0xB1,
 0xC0, 0xC7, 0x1A, 0x95, 0x0B, 0x62, 0x46, 0xB5, 0x88, 0x3E, 0x6C, 0xD8, 0xA7, 0x42, 0xCA, 0x61,
 0x9F, 0xD2, 0x00, 0xD2, 0xE8, 0xC8, 0xD9, 0x17, 0xE1, 0xA5, 0x4A, 0x98, 0x87, 0x19, 0xFB, 0xCC,
 0xD1, 0x55, 0xBD, 0x10, 0xBB, 0x68, 0x50, 0x84, 0x21, 0x94, 0x59, 0xAB, 0x06, 0x43, 0x97, 0x33,
 0xA3, 0x6E, 0xB0, 0x2C, 0x79, 0xAC, 0xA7, 0x91, 0x4C, 0xF5, 0x90, 0xB3, 0x8F, 0xEA, 0x14, 0x1B,
 0xA8, 0xAB, 0x57, 0x35, 0xC9, 0x1D, 0xE2, 0x8F, 0xCE, 0x87, 0x84, 0x26, 0xB9, 0x59, 0x84, 0x7D,
 0xAA, 0x67, 0x92, 0x84, 0x59, 0x69, 0x36, 0xD1, 0x30, 0xCE, 0xCF, 0x28, 0x6B, 0x7C, 0xF3, 0xA2,
 0x5B, 0xC3, 0xB5, 0x58, 0x84, 0xAA, 0xD9, 0x45, 0xC3, 0xB8, 0x48, 0x86, 0xD2, 0x2C, 0xC3, 0x3E,
 0x37, 0x15, 0xFA, 0x8D, 0x7B, 0xE3, 0x90, 0x52, 0xCF, 0x0D, 0x76, 0x2A, 0x51, 0x79, 0x71, 0xF6,
 0x57, 0x18, 0x50, 0x32, 0xB9, 0xAE, 0xCB, 0x90, 0x86, 0x38, 0x5B, 0x20, 0x68, 0x21, 0xC7, 0x98,
 0xAE, 0x30, 0x2E, 0x6E, 0x37, 0x5C, 0xB4, 0x84, 0xBC, 0x33, 0x9D, 0x3A, 0x3A, 0xDF, 0xB3, 0x42,
 0x3F, 0x60, 0x7D, 0xDB, 0xC2, 0x23, 0xC0, 0xD8, 0x5F, 0x9F, 0x38, 0x1D, 0x83, 0x15, 0x27, 0xAA,
 0x5B, 0x63, 0xCD, 0x5C, 0x5E, 0x48, 0x99, 0x8D, 0xB5, 0x48, 0x78, 0xA0, 0x0E, 0xA1, 0xD7, 0xDA,
 0x73, 0x32, 0x12, 0x35, 0x67, 0xA2, 0x10, 0x2C, 0x2C, 0x0B, 0x69, 0xB9, 0xFA, 0xD6, 0x0C, 0x5B,
 0x1F, 0xB1, 0xFD, 0x6D, 0x69, 0x1B, 0x56, 0xD6, 0x1E, 0x36, 0x88, 0xBB, 0x08, 0x69, 0x9D, 0xB5,
 0x53, 0x8B, 0x5B, 0xC1, 0x9C, 0x3B, 0x64, 0xA4, 0x62, 0xBB, 0x5D, 0xD4, 0x54, 0x9C, 0x2D, 0xAE,
 0x8A, 0x8D, 0xA0, 0x0A, 0x3B, 0x72, 0xD0, 0x18, 0x3B, 0x45, 0x22, 0xCB, 0x60, 0xC8, 0x49, 0xBB,
 0x32, 0x57, 0xE5, 0xF7, 0x6E, 0x5C, 0xB2, 0xA4, 0x78, 0x75, 0x1F, 0x7E, 0x53, 0xD9, 0x8E, 0xFC,
 0xF8, 0x24, 0xF5, 0x53, 0x80, 0x1D, 0x08, 0xB0, 0xBC, 0xD6, 0x1B, 0xC6, 0xAC, 0x40, 0x86, 0xC2,
 0x09, 0x7C, 0xE4, 0x4E, 0x31, 0xE4, 0x82, 0xAB, 0x93, 0xE8, 0xB0, 0x78, 0x61, 0x50, 0x49, 0x7D,
 0x96, 0xAA, 0xCF, 0x8A, 0x17, 0x22, 0x22, 0x21, 0x6C, 0xD1, 0x8C, 0x28, 0xB0, 0x16, 0xCE, 0xDF,
 0xD2, 0x3A, 0x85, 0xE8, 0x47, 0xB4, 0x01, 0x93, 0x76, 0x29, 0x6D, 0x7F, 0x5F, 0x9A, 0x11, 0xA2,
 0x95, 0xDE, 0x64, 0x52, 0xB6, 0x56, 0x9C, 0x4C, 0x3A, 0xCD, 0x4E, 0xB7, 0xB4, 0x61, 0xD2, 0x6A,
 0x99, 0x59, 0x2F, 0x6A, 0x32, 0x46, 0x9C, 0x4D, 0xCA, 0x21, 0xE8, 0xCF, 0xBC, 0x25, 0xF6, 0x35,
 0x40, 0x64, 0xC4, 0xED, 0x3E, 0xEA, 0xDA, 0x15, 0xB8, 0x21, 0xC8, 0xF7, 0x4B, 0x5D, 0x36, 0x4D,
 0xB3, 0x6B, 0xB7, 0xAC, 0x76, 0xA1, 0x63, 0x0A, 0x76, 0x0D, 0xF0, 0x06, 0x34, 0x76, 0xB0, 0x5D,
 0x90, 0x9E, 0x6D, 0x3C, 0x41, 0xA1, 0x43, 0x4B, 0xEC, 0x8D, 0x9A, 0xEC, 0x4F, 0xD1, 0x8C, 0x3C,
 0xAE, 0xFE, 0x60, 0x1B, 0x1D, 0x43, 0x1E, 0x09, 0x7F, 0x6A, 0xE6, 0x8C, 0x6A, 0x27, 0x5A, 0x2C,
 0x30, 0x82, 0x51, 0x16, 0xCE, 0x5B, 0x92, 0x56, 0xEA, 0x99, 0xF5, 0x89, 0xAB, 0xD2, 0x42, 0xB4,
 0xD4, 0x15, 0xE3, 0x6E, 0x68, 0x23, 0x9D, 0xFB, 0x13, 0xCF, 0x0A, 0x75, 0x65, 0xBA, 0x9A, 0x4B,
 0xAD, 0xF3, 0xEB, 0x47, 0x26, 0x0B, 0x1C, 0xC2, 0x1D, 0x3B, 0x74, 0x5D, 0x86, 0x68, 0x9D, 0xFA,
 0xA0, 0xA6, 0x66, 0xA2, 0x6A, 0x86, 0xDB, 0x2A, 0x3A, 0x53, 0x86, 0xCD, 0xDB, 0x8C, 0xC9, 0x04,
 0xA0, 0x26, 0x51, 0xC4, 0x39, 0xC4, 0x08, 0x3C, 0x50, 0x2A, 0x62, 0xB5, 0x9B, 0x5D, 0xE8, 0x2C,
 0x9C, 0xEB, 0x1A, 0x83, 0x68, 0xB2, 0x16, 0x54, 0x31, 0x31, 0x9D, 0x3F, 0x1D, 0xA3, 0x5A, 0xF3,
 0xA4, 0x79, 0xD2, 0x81, 0xBF, 0x34, 0x0D, 0x7A, 0xB1, 0x73, 0x49, 0xF3, 0xE6, 0x78, 0x5E, 0x26,
 0xF9, 0x94, 0xEF, 0x93, 0xE4, 0xA5, 0xB1, 0x52, 0x2C, 0xAA, 0x47, 0x52, 0x7A, 0xC3, 0xA4, 0xD5,
 0x28, 0x29, 0x2C, 0x39, 0x2E, 0xBD, 0xB9, 0x23, 0x6A, 0xBC, 0x65, 0x53, 0x88, 0xE7, 0xDE, 0x3F,
 0x75, 0x51, 0x55, 0xFF, 0xEF, 0xBD, 0x5D, 0x31, 0xC5, 0x57, 0xED, 0xE9, 0x1B, 0xDB, 0x25, 0x38,
 0xB4, 0x6F, 0x34, 0xF3, 0x51, 0xAF, 0xCB, 0x7E, 0x06, 0x24, 0x74, 0x61, 0x51, 0xE5, 0xC3, 0xEA,
 0x2A, 0xB7, 0xE7, 0x51, 0xC6, 0x6C, 0x61, 0x83, 0x09, 0x71, 0x9C, 0xBA, 0xE3, 0xAD, 0xCA, 0x3B,
 0x91, 0x62, 0x4F, 0x5E, 0xF3, 0xD3, 0x72, 0x97, 0xDF, 0x56, 0xDA, 0x10, 0x32, 0xD7, 0xFF, 0x84,
 0xB4, 0x5F, 0x77, 0xC0, 0x15, 0x86, 0xC6, 0x76, 0x85, 0x62, 0x0B, 0x7F, 0xDC, 0x6D, 0xA2, 0x4A,
 0xAE, 0x24, 0x3A, 0xC1, 0xC2, 0xC5, 0x5C, 0xB0, 0x22, 0xD4, 0x9A, 0x6D, 0xB1, 0xA8, 0x5A, 0x78,
 0x01, 0x11, 0xD7, 0x68, 0x7C, 0xEC, 0x20, 0xD6, 0xC1, 0x6F, 0xB5, 0xE4, 0x2E, 0x5D, 0x98, 0xA8,
 0xE4, 0x55, 0x34, 0xE1, 0xA6, 0xFB, 0x72, 0xB6, 0x4B, 0x1A, 0xA2, 0x77, 0xC8, 0xCF, 0xD5, 0x7A,
 0xB7, 0x2E, 0x69, 0xF7, 0xD3, 0x91, 0xA1, 0x1F, 0xB4, 0x41, 0x46, 0x8F, 0x92, 0xF6, 0xD4, 0xC7,
```

```
388  0xD7, 0x15, 0x94, 0x39, 0x91, 0xFF, 0xF6, 0xC5, 0x86, 0xE8, 0xF6, 0x6B, 0x7F, 0x5E, 0x00, 0xA4,
389  0x17, 0x35, 0xBA, 0x41, 0x85, 0xA9, 0xF3, 0xA7, 0xAC, 0xE2, 0x8F, 0xF1, 0x76, 0x9F, 0x69, 0x56,
390  0x48, 0x37, 0x05, 0x25, 0x54, 0xEF, 0xAA, 0x51, 0xF5, 0xD5, 0x9E, 0x74, 0xF0, 0x84, 0xE6, 0x5C,
391  0xCD, 0xE0, 0x7D, 0x6A, 0xA7, 0x38, 0xBB, 0xD5, 0x95, 0x7D, 0x82, 0xD2, 0xCC, 0x11, 0xEF, 0xCA,
392  0xE5, 0x7B, 0x9F, 0x96, 0x33, 0xCB, 0x9E, 0x1B, 0x33, 0xCF, 0x87, 0x24, 0x6A, 0x9F, 0x39, 0xCC,
393  0x30, 0x66, 0x2E, 0x4B, 0x3E, 0xC0, 0x83, 0x7F, 0xAF, 0xB5, 0x7B, 0xDA, 0x8B, 0x05, 0x05, 0x83,
394  0x8B, 0x44, 0xCB, 0xDD, 0xD6, 0x5A, 0x2F, 0x59, 0xB9, 0x0B, 0x64, 0x35, 0x17, 0x69, 0x81, 0x2A,
395  0x8E, 0xCA, 0xA2, 0x0C, 0xB3, 0xBE, 0x47, 0x53, 0xE8, 0xEC, 0x64, 0x8E, 0xA0, 0xED, 0x65, 0xEE,
396  0x8A, 0x80, 0xA3, 0x0E, 0xBF, 0x2A, 0xEE, 0xAE, 0x6C, 0x1A, 0xB6, 0x7A, 0xCD, 0x92, 0x29, 0x2D,
397  0xC7, 0x0B, 0x8A, 0xE3, 0x0A, 0x8D, 0xC1, 0x7E, 0x21, 0xD5, 0x4C, 0x24, 0xB7, 0x2E, 0xB5, 0x3B,
398  0x4F, 0xDC, 0xB9, 0xB5, 0x67, 0x2A, 0x95, 0xEE, 0xC2, 0x98, 0x2A, 0x0E, 0xC7, 0x8C, 0xCD, 0x5B,
399  0x4D, 0x6D, 0xA6, 0x2D, 0xDC, 0x7F, 0xA3, 0xF8, 0x0A, 0xD6, 0x9B, 0xEC, 0x82, 0x5C, 0xDF, 0xB0,
400  0xB0, 0x3E, 0x8D, 0xA6, 0x8A, 0x5C, 0xAB, 0xCA, 0x26, 0x60, 0x21, 0x0E, 0x33, 0x62, 0xDB, 0xB8,
401  0x70, 0x97, 0x93, 0xAD, 0x79, 0x2B, 0x36, 0x0F, 0x4C, 0x7E, 0xDD, 0xA6, 0xD4, 0xAD, 0x04, 0x45,
402  0xE1, 0x75, 0xFA, 0xD6, 0x6D, 0x47, 0x8C, 0x2C, 0x34, 0x79, 0x7B, 0xC4, 0xE9, 0x56, 0xA4, 0x50,
403  0x54, 0x6D, 0x70, 0xC7, 0xDB, 0xC4, 0xCC, 0x64, 0x60, 0x07, 0x36, 0x6A, 0x3D, 0x9B, 0x2B, 0x52,
404  0x0D, 0x4E, 0x95, 0x7B, 0x89, 0x06, 0xA7, 0xC9, 0x6D, 0x4F, 0x03, 0x76, 0x43, 0x91, 0x7A, 0xCB,
405  0x91, 0xB8, 0xDE, 0x65, 0x58, 0x0E, 0x0A, 0x82, 0xA1, 0xC9, 0x6E, 0x8C, 0x31, 0xD3, 0x77, 0x20,
406  0x0D, 0x6C, 0xB2, 0x34, 0x88, 0x3D, 0x34, 0x1D, 0x6F, 0xEA, 0x65, 0xCE, 0xF1, 0xF3, 0xE2, 0x0A,
407  0x04, 0x24, 0xCD, 0xA1, 0x99, 0xBA, 0x3A, 0x63, 0x72, 0xAA, 0xE4, 0x27, 0x73, 0xF4, 0xE0, 0xDE,
408  0xA3, 0x87, 0x0F, 0x7B, 0xDF, 0x3D, 0x70, 0xC7, 0xC1, 0x42, 0xFE, 0xFD, 0x8B, 0xB8, 0x98, 0x25,
409  0xEE, 0x88, 0x82, 0x3C, 0x4A, 0x29, 0xE8, 0x19, 0x0C, 0x4E, 0x39, 0xD3, 0x8C, 0x20, 0xA7, 0x20,
410  0x49, 0x8E, 0x6C, 0xB2, 0xB6, 0xEA, 0xC4, 0x8B, 0x86, 0x04, 0x50, 0x2E, 0xC6, 0xC8, 0xD7, 0x0C,
411  0xE1, 0xC3, 0x44, 0xE7, 0xC6, 0xFD, 0xD6, 0xE4, 0x35, 0x66, 0xEC, 0x5D, 0x65, 0x35, 0xE0, 0x4A,
412  0xC9, 0x02, 0x24, 0x47, 0x61, 0x3B, 0x8F, 0x21, 0x90, 0x71, 0x72, 0x76, 0x69, 0x2A, 0x67, 0x4C,
413  0x2C, 0x9F, 0xB4, 0xBE, 0x72, 0xA5, 0x44, 0x4C, 0x3D, 0xF1, 0xD1, 0x1C, 0x33, 0xF7, 0x97, 0x3F,
414  0xE6, 0xB3, 0xC9, 0x22, 0x11, 0x53, 0x9A, 0xA3, 0x37, 0x98, 0x67, 0x4E, 0x40, 0x59, 0x6B, 0xD6,
415  0x35, 0x2E, 0xB2, 0x98, 0xA5, 0xE6, 0x37, 0x23, 0x11, 0xE5, 0xE6, 0x75, 0x1D, 0x71, 0xB7, 0x29,
416  0x11, 0x88, 0xB3, 0xF3, 0x16, 0xDC, 0xC1, 0x96, 0xC8, 0x09, 0xC1, 0xB4, 0xAD, 0x96, 0x39, 0xFA,
417  0xF9, 0xF7, 0xEF, 0x9F, 0xD4, 0xDA, 0xCD, 0xEE, 0xF9, 0x55, 0xEB, 0xAC, 0xD7, 0x3D, 0x1E, 0x9C,
418  0x8A, 0x21, 0x9B, 0xF3, 0x6A, 0x9A, 0xA3, 0x5F, 0x19, 0x2F, 0xA8, 0x2F, 0xCD, 0xAB, 0x56, 0xBB,
419  0xD9, 0xDC, 0x9E, 0xD7, 0x23, 0x73, 0xF4, 0x96, 0xB3, 0x6A, 0x9F, 0x03, 0xAB, 0x66, 0x7B, 0x07,
420  0xB1, 0xCE, 0xCD, 0x11, 0xE7, 0x04, 0x4C, 0xAE, 0x1E, 0xF6, 0xCE, 0xB7, 0x67, 0xF4, 0x10, 0x64,
421  0x7A, 0x07, 0x9C, 0xCE, 0x41, 0xBB, 0xDE, 0x2E, 0xCA, 0xF5, 0xCC, 0x11, 0xE3, 0xD3, 0xEB, 0x36,
422  0xAF, 0xBA, 0xE7, 0x3B, 0xF0, 0x39, 0x33, 0x65, 0xA7, 0xC3, 0xDC, 0x3F, 0x3A, 0x32, 0x47, 0x17,
423  0x3F, 0x3C, 0xAF, 0x75, 0x41, 0xC6, 0xF6, 0xA3, 0xDE, 0xF6, 0xBC, 0xBB, 0xE0, 0x17, 0x4C, 0xC8,
424  0x4E, 0x1B, 0x18, 0x75, 0x77, 0x10, 0xB2, 0x63, 0x8E, 0x5E, 0x70, 0x4E, 0xC0, 0xE5, 0xAA, 0xF5,
425  0x70, 0x07, 0x91, 0xC0, 0xBD, 0x7E, 0xE6, 0x9C, 0xC0, 0xBF, 0x98, 0x7B, 0x55, 0xE4, 0x04, 0xB9,
426  0x97, 0x9B, 0xA6, 0x20, 0xE6, 0xD7, 0x33, 0x59, 0xEA, 0x74, 0x51, 0x4A, 0xF8, 0x3B, 0x84, 0x8E,
427  0x80, 0x5E, 0x6F, 0x9C, 0x10, 0x24, 0x1D, 0xA8, 0x24, 0x0E, 0xAA, 0xE5, 0x02, 0x45, 0x92, 0xF8,
428  0x6A, 0xAB, 0x39, 0xEA, 0x96, 0x28, 0xC0, 0x49, 0xD5, 0x84, 0xCA, 0x69, 0x53, 0xF2, 0x9B, 0xAC,
429  0x3F, 0x64, 0xA8, 0xB3, 0xFB, 0x79, 0xC0, 0x43, 0x3B, 0xA6, 0x12, 0xD5, 0x5B, 0x25, 0x1B, 0x8D,
430  0xAC, 0xE8, 0xCA, 0x1C, 0xF5, 0x3A, 0x65, 0xD6, 0xDE, 0x01, 0x8C, 0x31, 0xEF, 0x3D, 0x5D, 0x1C,
431  0x04, 0x1B, 0xE3, 0x91, 0x90, 0x9A, 0xA3, 0xA7, 0xF1, 0xF1, 0x2E, 0xA8, 0xD4, 0xCB, 0x34, 0xE5,
432  0xB4, 0x39, 0xB0, 0x28, 0xE2, 0x08, 0x64, 0xEA, 0x1D, 0x09, 0x4D, 0x82, 0xCC, 0xE7, 0x05, 0xE6,
433  0x36, 0x71, 0x61, 0xED, 0x80, 0x8F, 0x02, 0xBA, 0x31, 0x2A, 0x11, 0x21, 0x24, 0x35, 0x79, 0x74,
434  0x30, 0x44, 0x62, 0x51, 0xBE, 0x02, 0x3C, 0x02, 0x44, 0x43, 0x9F, 0xDF, 0xE5, 0xB8, 0x31, 0x22,
435  0x09, 0x29, 0x54, 0xC3, 0xF8, 0x78, 0x27, 0x54, 0x76, 0x49, 0x5F, 0x8A, 0x38, 0x12, 0x97, 0x28,
436  0x85, 0x75, 0x6F, 0x09, 0x97, 0x32, 0x69, 0x77, 0xC2, 0x65, 0x86, 0xFC, 0xC5, 0x56, 0xE9, 0x2B,
437  0xA6, 0x04, 0x54, 0xA2, 0xC3, 0x83, 0x85, 0x4A, 0x22, 0xCC, 0x57, 0x10, 0x2B, 0xB0, 0xFE, 0xF6,
438  0x48, 0xB0, 0x79, 0xC7, 0x2F, 0xE9, 0xCC, 0xD1, 0x33, 0x5C, 0x7F, 0xCD, 0x8E, 0x76, 0x81, 0xE3,
439  0x49, 0x48, 0xBD, 0x1D, 0x00, 0x89, 0x64, 0x11, 0x70, 0x34, 0x25, 0x1A, 0xE7, 0xB7, 0x84, 0xC6,
440  0xF9, 0x2D, 0xA2, 0x81, 0xF0, 0x7B, 0x07, 0x2F, 0xB1, 0xB3, 0x31, 0x1C, 0x11, 0xA1, 0x39, 0xBA,
441  0xBC, 0x5A, 0x78, 0x01, 0xBB, 0x5B, 0xF8, 0x25, 0xFB, 0xBE, 0x53, 0x90, 0x9C, 0xED, 0x80, 0x49,
442  0x2C, 0x90, 0x8C, 0x91, 0x33, 0x89, 0xCA, 0xD9, 0x2D, 0xA1, 0x52, 0x26, 0xEB, 0x2E, 0xA8, 0x4C,
443  0x11, 0x71, 0x2D, 0x4C, 0x1C, 0x76, 0xE7, 0xE2, 0xA6, 0xC0, 0x28, 0xB4, 0xE6, 0xE8, 0xFB, 0xE4,
444  0xCB, 0x2E, 0xC0, 0x34, 0x77, 0xC0, 0x45, 0x95, 0x27, 0x1D, 0x2F, 0x67, 0xB0, 0x58, 0xBE, 0x25,
445  0x6C, 0x5A, 0xAD, 0xDB, 0xAC, 0x2A, 0x0B, 0x6C, 0x11, 0xE4, 0xBC, 0xC7, 0x93, 0x09, 0x2C, 0x83,
446  0x36, 0x2F, 0x2D, 0x29, 0x72, 0xA8, 0x2F, 0xE2, 0xBB, 0x71, 0xC9, 0xBF, 0x6F, 0xBC, 0x87, 0x91,
447  0x61, 0xF7, 0xB9, 0x36, 0x32, 0x9A, 0xFA, 0xB5, 0xF0, 0x6B, 0x2F, 0x96, 0x73, 0xDB, 0x5D, 0x0D,
448  0x60, 0x82, 0xA7, 0x7C, 0x53, 0x7D, 0x6B, 0x1E, 0x6D, 0xF0, 0x6C, 0x1F, 0x5D, 0xF3, 0xC7, 0x10,
449  0x77, 0x59, 0x48, 0xBF, 0xC1, 0xB6, 0xF1, 0x0B, 0x71, 0xB7, 0x57, 0xA6, 0xCB, 0x04, 0xC1, 0xD8,
450  0xDD, 0x8D, 0xCB, 0x19, 0x2C, 0x91, 0xE0, 0x60, 0x37, 0x26, 0x3D, 0xF0, 0x24, 0xBC, 0x20, 0xE8,
451  0x4B, 0x58, 0xC4, 0xA3, 0xD5, 0x78, 0xF3, 0x82, 0xB2, 0x1A, 0x43, 0x5D, 0xFE, 0xED, 0xA9, 0x71,
452  0xC9, 0x6F, 0x03, 0xDB, 0x38, 0x5D, 0x89, 0x2B, 0xD4, 0x55, 0x1C, 0x5D, 0x24, 0x2A, 0x29, 0xA7,
453  0xB9, 0xB6, 0x27, 0xAA, 0x0F, 0xA0, 0xAA, 0xFB, 0xA2, 0x1A, 0xF5, 0x22, 0x01, 0xF9, 0x05, 0x3D,
454  0x53, 0xD1, 0xB6, 0x9A, 0x8E, 0xB7, 0xD8, 0x8A, 0x59, 0xAB, 0xCD, 0xDB, 0x30, 0x6B, 0x05, 0x30,
455  0xD9, 0x4B, 0x76, 0x87, 0xA0, 0x6D, 0x00, 0x5E, 0x7B, 0x01, 0x8A, 0xCD, 0x7A, 0x18, 0xA0, 0xB8,
456  0xBE, 0x87, 0x06, 0x0A, 0xBC, 0xE5, 0x3D, 0xAB, 0xA3, 0xDB, 0x04, 0x15, 0x27, 0x34, 0x47, 0xAF,
457  0x90, 0x1B, 0x42, 0x91, 0xD9, 0x17, 0x60, 0xF1, 0xC4, 0x07, 0x0B, 0x2F, 0xA9, 0xF7, 0xA1, 0xA1,
458  0x03, 0x41, 0xE6, 0x9E, 0xBD, 0xF9, 0x72, 0x47, 0xD2, 0x89, 0x94, 0xF8, 0x0A, 0x8E, 0x36, 0x6E,
459  0x0C, 0x22, 0x0E, 0xB7, 0xDC, 0x11, 0x88, 0xA5, 0xD4, 0xF6, 0xCD, 0xC0, 0xDB, 0xD0, 0x75, 0xAF,
460  0x77, 0xE9, 0x04, 0x2E, 0x1C, 0x2F, 0xB4, 0xB7, 0xE7, 0x00, 0x6D, 0xC0, 0x8F, 0x93, 0x09, 0xB1,
461  0xB6, 0x6F, 0x24, 0xA0, 0x09, 0x78, 0xE1, 0xCD, 0x2B, 0xD2, 0xDF, 0x72, 0xE1, 0xC5, 0xD6, 0x16,
462  0x2B, 0x39, 0x0B, 0x50, 0xBC, 0xBC, 0xD8, 0x6B, 0xE1, 0x85, 0x39, 0x0F, 0x94, 0x19, 0x98, 0xB6,
463  0x87, 0x4E, 0x0A, 0x20, 0xC4, 0x7B, 0xEE, 0x3C, 0xDB, 0x80, 0x25, 0x28, 0xE3, 0x8C, 0x1E, 0x2D,
464  0xBF, 0x0F, 0xB5, 0xBE, 0x4B, 0x24, 0x4A, 0xAF, 0xEE, 0x5A, 0x67, 0x9D, 0x5E, 0xBC, 0xBC, 0xEB,
465  0xB4, 0x3F, 0xEF, 0x02, 0x8F, 0x31, 0xBF, 0x5D, 0x7C, 0xDA, 0xDB, 0x40, 0x03, 0xD9, 0xE8, 0x35,
466  0xBB, 0xCE, 0xB0, 0x41, 0xC2, 0xDE, 0x3D, 0x90, 0xDA, 0x87, 0x8B, 0xA4, 0xF6, 0x17, 0x10, 0x4A,
467  0xD3, 0x2D, 0x32, 0xDE, 0x94, 0x65, 0xBC, 0xEF, 0x2F, 0xF6, 0x83, 0xD0, 0xF4, 0x60, 0xA9, 0x6E,
468  0x7A, 0xD0, 0x54, 0x67, 0x88, 0x9B, 0xAD, 0x62, 0x98, 0xB6, 0xEC, 0x60, 0x25, 0xA1, 0xD8, 0xCB,
469  0xDA, 0x25, 0xC9, 0xB5, 0xAE, 0x76, 0xC9, 0x72, 0x91, 0x18, 0xE9, 0x24, 0xD7, 0x4B, 0xAE, 0x8A,
470  0x9C, 0x7D, 0xDE, 0xCB, 0xBA, 0xDD, 0x32, 0x69, 0x77, 0x09, 0x1A, 0x1F, 0xAD, 0xDE, 0x4F, 0xE7,
471  0x68, 0x63, 0x30, 0x24, 0x1D, 0x60, 0xF1, 0xEA, 0xC9, 0x3E, 0xDB, 0x85, 0x68, 0xDE, 0xC3, 0xC4,
472  0x51, 0xAC, 0xF5, 0xA1, 0x73, 0x9D, 0x83, 0xDD, 0xCD, 0x93, 0x1D, 0x23, 0x32, 0x47, 0x2F, 0xB1,
473  0x1B, 0x18, 0x17, 0x9E, 0x2F, 0xDF, 0xFD, 0xB4, 0x17, 0xD4, 0xF8, 0xCC, 0x87, 0x81, 0x4C, 0x28,
474  0x7D, 0x68, 0xBC, 0x66, 0x73, 0xE2, 0xFB, 0x9E, 0xBF, 0x31, 0x64, 0x92, 0x0E, 0x96, 0x15, 0xF5,
```

```
475  0x57, 0xFC, 0x68, 0x2F, 0x70, 0x45, 0xB3, 0x1E, 0x06, 0xB1, 0x58, 0xE7, 0x43, 0x83, 0xB6, 0x9C,
476  0x38, 0x64, 0xB1, 0x31, 0x64, 0x9C, 0xCA, 0x1C, 0xBD, 0xAB, 0x3F, 0x87, 0x7F, 0xF7, 0x02, 0x97,
477  0x98, 0xF1, 0x30, 0x60, 0x49, 0x6D, 0x0F, 0x0D, 0xD5, 0x78, 0xB1, 0x79, 0x3A, 0x04, 0x1A, 0x73,
478  0xF4, 0xF4, 0xA7, 0xFD, 0xF4, 0x7E, 0x6C, 0xB2, 0x8A, 0x08, 0xED, 0x84, 0x07, 0x57, 0xEA, 0xD0,
479  0x68, 0xAC, 0xB6, 0x40, 0x63, 0xC5, 0x04, 0xFF, 0x6D, 0x4F, 0x68, 0xAC, 0xAA, 0xA3, 0xF1, 0x99,
480  0xE3, 0x65, 0xF5, 0x25, 0xE0, 0xC3, 0x9F, 0xC5, 0x18, 0xA3, 0xCD, 0xCB, 0x51, 0x44, 0xC8, 0x6E,
481  0x1A, 0x83, 0x23, 0xE3, 0x29, 0xDA, 0x4F, 0x41, 0x8A, 0xE7, 0xDD, 0x47, 0x08, 0x25, 0x4A, 0x1E,
482  0x1A, 0xA7, 0x09, 0xB2, 0xF0, 0x7B, 0x1B, 0xD3, 0x6D, 0xAE, 0x2D, 0x2B, 0xB4, 0xE6, 0xE8, 0x39,
483  0x7C, 0x31, 0x9E, 0xF1, 0x2F, 0xFB, 0x6A, 0xF9, 0xD4, 0xF9, 0xF7, 0x81, 0x5A, 0x4A, 0xDF, 0x2F,
484  0x02, 0x38, 0x68, 0xB0, 0xBD, 0xA9, 0xBB, 0xD5, 0x23, 0x0D, 0x29, 0x72, 0x09, 0xDF, 0x1B, 0xF1,
485  0x7D, 0xBF, 0x00, 0x26, 0x42, 0xEC, 0x0D, 0x43, 0x45, 0xEF, 0x7D, 0xC0, 0x18, 0x3D, 0x16, 0xC4,
486  0x8B, 0xB4, 0x78, 0x15, 0x5E, 0x19, 0x52, 0xF2, 0xE1, 0x27, 0x7E, 0x4B, 0x0B, 0xA6, 0xF5, 0x80,
487  0x12, 0xC7, 0x81, 0x85, 0x30, 0xA6, 0xC6, 0x5B, 0x76, 0x38, 0x38, 0x15, 0x03, 0xAA, 0x73, 0x91,
488  0xCF, 0xDC, 0xB0, 0x97, 0x50, 0xA2, 0xB9, 0x39, 0x7A, 0xCB, 0x5E, 0x12, 0x08, 0xBC, 0xD8, 0xB7,
489  0xCD, 0x99, 0x71, 0x23, 0x62, 0xD7, 0xF7, 0x40, 0xA8, 0x18, 0x24, 0xF9, 0xAE, 0x26, 0xD3, 0x88,
490  0x8E, 0x94, 0xDF, 0x46, 0x97, 0x7C, 0xB0, 0xC1, 0xBC, 0xAC, 0x7C, 0x3A, 0x76, 0xD5, 0xC2, 0xCA,
491  0xBF, 0xB8, 0x31, 0x38, 0x75, 0x91, 0xC6, 0xDC, 0x39, 0x28, 0x0C, 0xC4, 0xDB, 0x25, 0x73, 0x58,
492  0xC5, 0xCF, 0x33, 0x71, 0x4B, 0x24, 0x8F, 0x69, 0xC6, 0x6A, 0x65, 0x1F, 0xDF, 0x94, 0xDB, 0x4C,
493  0xD5, 0x82, 0x96, 0x3F, 0x88, 0x29, 0xEB, 0x21, 0x3B, 0x8C, 0xCD, 0xFF, 0x9F, 0x7F, 0x97, 0xF9,
494  0x0C, 0x7B, 0xF7, 0x67, 0x22, 0x98, 0x69, 0x04, 0xBE, 0x35, 0x34, 0xF3, 0x9E, 0x8E, 0xCA, 0xD1,
495  0xFC, 0x54, 0xA7, 0x7A, 0x66, 0xB0, 0xC6, 0xD6, 0x83, 0xC0, 0xF2, 0xC9, 0x82, 0x8E, 0xEE, 0xD8,
496  0x9E, 0x15, 0xCE, 0xB1, 0x4B, 0x1B, 0xC8, 0xB6, 0x2F, 0x97, 0x70, 0xF0, 0x92, 0x04, 0x14, 0x83,
497  0x15, 0x6A, 0x47, 0xCF, 0x7E, 0x7C, 0x75, 0x21, 0x9E, 0x12, 0x7B, 0xE9, 0x21, 0x1B, 0xDB, 0x47,
498  0x27, 0xC6, 0x24, 0x74, 0x85, 0x9B, 0xD7, 0x30, 0x1B, 0x2B, 0xDE, 0xBB, 0xBA, 0x44, 0xBE, 0x31,
499  0x46, 0x01, 0x7E, 0xE1, 0x05, 0xD4, 0x18, 0x1A, 0x31, 0x47, 0xC7, 0xB3, 0xF8, 0x7D, 0xBF, 0x0D,
500  0xCF, 0x27, 0x53, 0xE2, 0xCA, 0x91, 0x42, 0xD9, 0x5F, 0x7D, 0x07, 0x86, 0xC6, 0x54, 0xDF, 0x1A,
501  0x47, 0xFD, 0xF3, 0xD6, 0x11, 0x7B, 0x1C, 0x0F, 0x60, 0x80, 0x1F, 0x00, 0x02, 0x0C, 0x03, 0x20,
502  0xC0, 0x87, 0x23, 0xF9, 0x78, 0x20, 0x76, 0x1A, 0xDC, 0xE4, 0x4C, 0x40, 0x26, 0x6D, 0xED, 0x48,
503  0xE0, 0x74, 0xC4, 0x1E, 0x34, 0xBE, 0x89, 0x29, 0x83, 0x99, 0xB7, 0x2A, 0xA2, 0xF4, 0xF1, 0xDC,
504  0x5B, 0xE2, 0x0C, 0x71, 0x4C, 0x2D, 0xBD, 0xB9, 0x74, 0xEA, 0xC8, 0xEB, 0x8F, 0x8E, 0xA3, 0x01,
505  0xF1, 0x7B, 0xCC, 0x86, 0x06, 0xF5, 0x43, 0x9C, 0x66, 0x8B, 0xDD, 0x32, 0xAE, 0x91, 0x58, 0x85,
506  0x8C, 0x27, 0xC8, 0x09, 0x32, 0x9C, 0xC3, 0x85, 0x8D, 0x28, 0x7E, 0xC7, 0x76, 0x0C, 0x61, 0x40,
507  0x0D, 0x3B, 0x27, 0x62, 0xFB, 0xF0, 0x44, 0x9E, 0x79, 0x03, 0x7C, 0x29, 0x3E, 0x4E, 0x66, 0x55,
508  0x7F, 0x06, 0x8A, 0xF4, 0xD7, 0xA1, 0xE1, 0x86, 0x10, 0xC2, 0x8F, 0xB9, 0x0A, 0x46, 0x3F, 0x75,
509  0x96, 0x53, 0x3B, 0x90, 0x9D, 0xE4, 0x3B, 0xDB, 0xF9, 0x9C, 0xFC, 0x47, 0x32, 0x61, 0x13, 0x37,
510  0xF8, 0x1B, 0xE4, 0x87, 0xC0, 0xE3, 0x28, 0xCA, 0xEE, 0x47, 0xC9, 0x8B, 0x79, 0x55, 0x22, 0x6E,
511  0x87, 0x86, 0xEC, 0x83, 0xE5, 0xF9, 0xA5, 0x3C, 0x71, 0xF7, 0xEE, 0x32, 0xE6, 0x6B, 0x28, 0xC3,
512  0xE0, 0x54, 0x72, 0xE2, 0x06, 0x4E, 0x28, 0x4F, 0x3F, 0xAF, 0xF3, 0xCE, 0xF0, 0x88, 0x98, 0x2B,
513  0x1C, 0xEE, 0xC4, 0x92, 0xA7, 0x2C, 0xF0, 0xE0, 0x41, 0x9A, 0xDB, 0xDD, 0xA1, 0xA4, 0x4A, 0x34,
514  0x11, 0xE3, 0x21, 0x32, 0x20, 0xF2, 0x40, 0x6D, 0xF9, 0x4C, 0xBC, 0x14, 0x89, 0x4C, 0x6A, 0x77,
515  0x53, 0x86, 0x8F, 0x65, 0x9C, 0x30, 0x13, 0x11, 0x9B, 0x1B, 0x88, 0x5F, 0x33, 0x3C, 0x4E, 0x9E,
516  0x7A, 0x15, 0xF2, 0x3D, 0xE6, 0x5E, 0x5F, 0xC3, 0xF2, 0xF2, 0xDB, 0x31, 0xD8, 0x9F, 0x39, 0x73,
517  0xF2, 0x83, 0x1C, 0x9F, 0x4C, 0xA5, 0x72, 0x9C, 0xA6, 0x38, 0x32, 0xC5, 0x32, 0x72, 0xB3, 0x0F,
518  0x9F, 0x00, 0x86, 0xB2, 0x9D, 0xEF, 0xE4, 0xF9, 0xFC, 0x8C, 0x39, 0xD9, 0x87, 0x4F, 0xBC, 0x3E,
519  0xB0, 0x50, 0x82, 0xE8, 0x0E, 0x09, 0x8D, 0x62, 0x9C, 0xDD, 0x6A, 0xCC, 0x54, 0xE2, 0x22, 0xC0,
520  0x61, 0x11, 0xAB, 0x4C, 0x01, 0xD7, 0x30, 0x14, 0x01, 0x55, 0x13, 0xF5, 0xE9, 0x29, 0xAF, 0x35,
521  0x8C, 0xB9, 0x8C, 0x95, 0xF4, 0xEF, 0x77, 0x54, 0xE1, 0x6F, 0xA2, 0xF0, 0x89, 0x53, 0x99, 0x8A,
522  0x27, 0xF3, 0xE3, 0xC8, 0x62, 0xCC, 0xD5, 0x13, 0x87, 0x91, 0xAF, 0x2B, 0x89, 0xFC, 0x3C, 0x31,
523  0xAB, 0x05, 0x39, 0x4C, 0xF1, 0xF8, 0x7E, 0x46, 0x54, 0xD5, 0xD5, 0x41, 0xEE, 0x96, 0xA1, 0xBE,
524  0x80, 0x64, 0x0C, 0xA9, 0xF0, 0x63, 0x8A, 0x0F, 0xDF, 0xB0, 0x8F, 0x99, 0x88, 0xDF, 0xC4, 0xE5,
525  0xFD, 0xBA, 0xE7, 0x62, 0x3D, 0x77, 0xD5, 0xD9, 0x75, 0x3C, 0x45, 0x29, 0xCE, 0x32, 0x0D, 0xC7,
526  0x73, 0x42, 0x35, 0x0C, 0x8F, 0x20, 0x0D, 0xEB, 0x78, 0xC9, 0x06, 0x2D, 0x21, 0xF0, 0x31, 0x0D,
527  0x7D, 0x57, 0x8D, 0x26, 0x91, 0x91, 0xFE, 0x0E, 0xB1, 0x7F, 0x0D, 0x8C, 0x3E, 0xDC, 0xFF, 0x14,
528  0xE5, 0xF7, 0x9B, 0x53, 0xFE, 0x6C, 0x8E, 0xE7, 0x3C, 0x86, 0x0A, 0x30, 0xBC, 0xFF, 0x89, 0x43,
529  0x7D, 0xF3, 0x00, 0xA6, 0x84, 0x2F, 0x7C, 0xE2, 0x9B, 0x0F, 0x82, 0xC5, 0x84, 0xBD, 0x3E, 0xBB,
530  0xC6, 0x59, 0x44, 0xB8, 0x35, 0xE8, 0x0C, 0xBB, 0x35, 0x1F, 0x07, 0x0B, 0x60, 0x8F, 0x93, 0x44,
531  0x16, 0xCD, 0xE8, 0x39, 0x18, 0x4A, 0xCD, 0xB4, 0xF6, 0xC1, 0xC7, 0x40, 0x07, 0x02, 0x50, 0xCF,
532  0xB8, 0xFF, 0x89, 0xB3, 0xB8, 0x31, 0x26, 0x10, 0xCD, 0xC1, 0x0C, 0xDB, 0x27, 0x50, 0x77, 0x10,
533  0x65, 0x4F, 0xA6, 0xDF, 0xFF, 0x14, 0xB1, 0x6A, 0x88, 0x9F, 0x6E, 0x3E, 0xC4, 0x1E, 0x12, 0x17,
534  0x83, 0xA8, 0x86, 0xF1, 0x13, 0x0D, 0xCE, 0xEB, 0x2D, 0x47, 0xC1, 0xF3, 0x9F, 0x38, 0x4E, 0xED,
535  0x48, 0xBC, 0x7E, 0x41, 0xE6, 0xE8, 0x06, 0x34, 0x9D, 0x97, 0x08, 0xC4, 0x56, 0x93, 0x3B, 0xCF,
536  0x3B, 0x9E, 0x6B, 0x39, 0xC4, 0xFA, 0xC8, 0x12, 0xF3, 0x71, 0x5A, 0x70, 0x11, 0xE9, 0x4E, 0x43,
537  0xBC, 0x4E, 0xEB, 0xB5, 0x67, 0xE3, 0x8C, 0x9B, 0x1E, 0x33, 0x31, 0x4E, 0x4F, 0xC1, 0xCA, 0xC8,
538  0x8E, 0x52, 0x92, 0xC0, 0x88, 0xBD, 0x77, 0x45, 0x98, 0x29, 0x65, 0x61, 0xA1, 0x8C, 0xD4, 0x45,
539  0xD8, 0x2C, 0xA9, 0xD6, 0x91, 0xCA, 0x89, 0xDB, 0x0A, 0xF4, 0x8C, 0xD8, 0x16, 0x7F, 0x05, 0x9E,
540  0x5B, 0x3B, 0xBE, 0x13, 0x9B, 0x61, 0x9D, 0x07, 0x9B, 0x40, 0x61, 0x90, 0x32, 0x51, 0x9E, 0x99,
541  0xD2, 0x5D, 0xFD, 0x51, 0x92, 0x49, 0x72, 0x6C, 0x26, 0x3E, 0x4A, 0x4D, 0xE3, 0x05, 0x8D, 0xCF,
542  0xFC, 0x07, 0x77, 0x9A, 0x3F, 0x4F, 0x44, 0x11, 0x54, 0x72, 0xD2, 0xB1, 0x62, 0x30, 0xE1, 0x81,
543  0xEC, 0xBF, 0x1E, 0x51, 0x1B, 0x11, 0xE8, 0xAE, 0x2F, 0x1D, 0xCC, 0x0E, 0x9F, 0x5E, 0xFF, 0x00,
544  0xC5, 0x5B, 0xB4, 0x20, 0x5C, 0x9A, 0x84, 0xE0, 0x22, 0x6E, 0xFF, 0x4A, 0x29, 0x93, 0x56, 0x51,
545  0xE1, 0xC1, 0xDB, 0x77, 0x91, 0x71, 0x8A, 0x38, 0xC4, 0x9D, 0x7E, 0x8A, 0x94, 0x71, 0x2D, 0xA7,
546  0x4D, 0xF5, 0xF7, 0x0A, 0xBD, 0x9A, 0xED, 0x8A, 0xE8, 0x95, 0x96, 0x5E, 0xA1, 0xE6, 0xAE, 0x5C,
547  0x4E, 0xAC, 0x36, 0xB7, 0x47, 0x8A, 0xB1, 0x03, 0xEA, 0x2D, 0xC4, 0x1A, 0x23, 0xE3, 0xE6, 0x2B,
548  0xE2, 0xDA, 0xDE, 0xAA, 0xC1, 0xCE, 0xD7, 0x64, 0x91, 0x54, 0x15, 0x6D, 0x10, 0x17, 0x0C, 0xF8,
549  0xE2, 0x97, 0x57, 0x2F, 0x59, 0xD2, 0x51, 0xD7, 0x2A, 0x47, 0xE9, 0x0E, 0x87, 0xBF, 0xEB, 0x5C,
550  0x3B, 0x03, 0x83, 0xAD, 0x01, 0x4D, 0xB3, 0x48, 0x36, 0x71, 0x63, 0xC9, 0x62, 0x81, 0x1D, 0x7E,
551  0x10, 0x73, 0xB2, 0xD2, 0x93, 0x02, 0xF8, 0xB8, 0x54, 0x16, 0x6F, 0x91, 0x15, 0x05, 0x22, 0xF1,
552  0x09, 0xA5, 0xE0, 0xB0, 0x86, 0x70, 0xE5, 0x80, 0x65, 0x19, 0xB9, 0xCE, 0xBB, 0x63, 0xA8, 0xE0,
553  0xE7, 0x04, 0x7D, 0x62, 0x26, 0x19, 0x65, 0x69, 0xE1, 0x95, 0x4C, 0x89, 0x16, 0x10, 0x99, 0xF8,
554  0xF1, 0x7B, 0x6B, 0x0C, 0xC9, 0xF1, 0x19, 0x78, 0x7E, 0xC3, 0x05, 0x0D, 0x8E, 0x6F, 0x8A, 0xD4,
555  0x11, 0xE6, 0x4A, 0x80, 0xAC, 0x2A, 0x04, 0x4F, 0x43, 0x7A, 0x6E, 0x29, 0xFB, 0xE8, 0xD9, 0xA9,
556  0xDE, 0x2B, 0xAE, 0xDD, 0xB2, 0x36, 0x2D, 0xCF, 0xB0, 0xC3, 0x75, 0xD3, 0x8A, 0x3E, 0x25, 0xC5,
557  0x20, 0x49, 0x30, 0x6B, 0xC2, 0x66, 0xDA, 0x14, 0xC5, 0x2F, 0xA2, 0x01, 0x91, 0xEC, 0x6A, 0x40,
558  0xE4, 0xC8, 0x9E, 0xEE, 0xE2, 0x32, 0xED, 0x42, 0x06, 0x72, 0x99, 0xC5, 0x0C, 0xF6, 0xD6, 0x8F,
559  0x19, 0x2B, 0xD0, 0xD2, 0x09, 0xAA, 0x14, 0x0A, 0x6D, 0x06, 0x2C, 0xAC, 0x18, 0x62, 0x86, 0x48,
560  0xDA, 0x6C, 0xB7, 0x99, 0xAE, 0x0E, 0x17, 0x21, 0x58, 0x69, 0x1E, 0xF9, 0xA4, 0xF8, 0x8D, 0xB5,
561  0x6C, 0x71, 0xF0, 0x40, 0x0B, 0x57, 0x14, 0xD4, 0x70, 0x5A, 0xC9, 0x04, 0xB2, 0xDF, 0x2B, 0x21,
```

```
562  0x50, 0xEE, 0xBA, 0xE0, 0xB4, 0xF0, 0xD3, 0xBA, 0xD8, 0x1A, 0x23, 0xC3, 0xB8, 0xE3, 0x18, 0x73,
563  0x46, 0x24, 0xBB, 0xA2, 0x04, 0xF1, 0xF5, 0xEE, 0x34, 0x0B, 0xF9, 0x5A, 0x57, 0x7A, 0xA3, 0xA0,
564  0x15, 0xDD, 0xB7, 0x96, 0xE8, 0x83, 0x8B, 0x95, 0xC7, 0xAA, 0xF2, 0x51, 0x97, 0x5D, 0x42, 0xA1,
565  0xDE, 0x65, 0x27, 0xD4, 0xC7, 0x15, 0xD5, 0xC7, 0x52, 0x7D, 0x46, 0x90, 0x34, 0x84, 0xE5, 0x2D,
566  0x7F, 0xEC, 0x8C, 0xBF, 0x3D, 0x4D, 0x34, 0x5B, 0x8D, 0x0B, 0xE5, 0x94, 0xAD, 0xB8, 0xA2, 0x5E,
567  0x31, 0x41, 0xEA, 0x9E, 0x62, 0xA1, 0xD6, 0x6A, 0x5C, 0x4D, 0xAD, 0xA8, 0x95, 0x67, 0x04, 0x89,
568  0x5A, 0xFA, 0x86, 0x3F, 0x52, 0x25, 0xDE, 0x42, 0xE6, 0xFF, 0xA7, 0x4B, 0xFC, 0xCE, 0x94, 0x58,
569  0x58, 0xB1, 0xFF, 0x5A, 0x5A, 0xCA, 0xC4, 0x30, 0x45, 0xC9, 0x78, 0xC9, 0x50, 0x4A, 0x1A, 0x8F,
570  0x54, 0xA8, 0x63, 0x39, 0x0A, 0xA9, 0xA3, 0x41, 0xA2, 0x06, 0xC6, 0x5F, 0x2B, 0x19, 0x2B, 0x1E,
571  0x9D, 0x04, 0x42, 0xC2, 0x40, 0x34, 0xE0, 0x23, 0xE3, 0x2C, 0xBB, 0xD4, 0x14, 0x8D, 0x90, 0x50,
572  0x36, 0xD3, 0xFE, 0xA8, 0x03, 0x62, 0x95, 0x52, 0x63, 0xE2, 0x00, 0x11, 0xF4, 0x79, 0x62, 0x96,
573  0x8A, 0x82, 0x1C, 0xEC, 0xD3, 0x9A, 0xF9, 0x93, 0x83, 0xD9, 0xF2, 0x41, 0xDE, 0x14, 0x7E, 0xF1,
574  0xC3, 0x73, 0xC3, 0xF3, 0x0D, 0xF1, 0x16, 0x4D, 0x3F, 0x7E, 0x6B, 0x8E, 0x21, 0x5F, 0x31, 0xC7,
575  0x17, 0x69, 0xC4, 0x9D, 0x1A, 0x74, 0x46, 0x02, 0xE8, 0x59, 0xD9, 0x93, 0xE0, 0xF8, 0xAE, 0x19,
576  0xBF, 0x45, 0xAE, 0x54, 0x3D, 0xD1, 0xA4, 0x7E, 0x17, 0x2B, 0x92, 0x31, 0xA7, 0xA0, 0x49, 0x6C,
577  0x79, 0x57, 0xEA, 0xB8, 0x96, 0x58, 0x8A, 0x96, 0x85, 0x1B, 0x98, 0x30, 0x3E, 0xFD, 0xC5, 0x5A,
578  0x51, 0xAF, 0x40, 0xA9, 0x21, 0x63, 0xB2, 0xC4, 0x96, 0x89, 0xAE, 0x6B, 0xD6, 0xD4, 0xAD, 0xBD,
579  0x0B, 0x10, 0x65, 0x5B, 0x49, 0xDA, 0x6C, 0x9E, 0x8F, 0x8A, 0xB0, 0xB8, 0xA8, 0x72, 0xE2, 0x33,
580  0x38, 0x8D, 0x36, 0x2C, 0xC5, 0x37, 0xF1, 0x52, 0xAE, 0xC1, 0xA9, 0xF8, 0x9F, 0x0A, 0xFF, 0x0B,
581  0x9B, 0xFC, 0x8E, 0x51, 0xC1, 0x70, 0x00, 0x00
582 };
```

5.27 WiFiTelemetry/camera_pins.h File Reference

5.28 camera_pins.h

Go to the documentation of this file.

```
1
2 #if defined(CAMERA_MODEL_WROVER_KIT)
3 #define PWDN_GPIO_NUM    -1
4 #define RESET_GPIO_NUM   -1
5 #define XCLK_GPIO_NUM    21
6 #define SIOD_GPIO_NUM    26
7 #define SIOC_GPIO_NUM    27
8
9 #define Y9_GPIO_NUM      35
10 #define Y8_GPIO_NUM      34
11 #define Y7_GPIO_NUM      39
12 #define Y6_GPIO_NUM      36
13 #define Y5_GPIO_NUM      19
14 #define Y4_GPIO_NUM      18
15 #define Y3_GPIO_NUM       5
16 #define Y2_GPIO_NUM       4
17 #define VSYNC_GPIO_NUM   25
18 #define HREF_GPIO_NUM    23
19 #define PCLK_GPIO_NUM    22
20
21 #elif defined(CAMERA_MODEL_ESP_EYE)
22 #define PWDN_GPIO_NUM    -1
23 #define RESET_GPIO_NUM   -1
24 #define XCLK_GPIO_NUM    4
25 #define SIOD_GPIO_NUM    18
26 #define SIOC_GPIO_NUM    23
27
28 #define Y9_GPIO_NUM      36
29 #define Y8_GPIO_NUM      37
30 #define Y7_GPIO_NUM      38
31 #define Y6_GPIO_NUM      39
32 #define Y5_GPIO_NUM      35
33 #define Y4_GPIO_NUM      14
34 #define Y3_GPIO_NUM      13
35 #define Y2_GPIO_NUM      34
36 #define VSYNC_GPIO_NUM   5
37 #define HREF_GPIO_NUM    27
38 #define PCLK_GPIO_NUM    25
39
40 #elif defined(CAMERA_MODEL_M5STACK_PSRAM)
41 #define PWDN_GPIO_NUM     -1
42 #define RESET_GPIO_NUM    15
43 #define XCLK_GPIO_NUM     27
44 #define SIOD_GPIO_NUM     25
45 #define SIOC_GPIO_NUM     23
46
47 #define Y9_GPIO_NUM       19
48 #define Y8_GPIO_NUM       36
49 #define Y7_GPIO_NUM       18
50 #define Y6_GPIO_NUM       39
51 #define Y5_GPIO_NUM        5
```

```
52 #define Y4_GPIO_NUM       34
53 #define Y3_GPIO_NUM       35
54 #define Y2_GPIO_NUM       32
55 #define VSYNC_GPIO_NUM    22
56 #define HREF_GPIO_NUM     26
57 #define PCLK_GPIO_NUM     21
58
59 #elif defined(CAMERA_MODEL_M5STACK_V2_PSRAM)
60 #define PWDN_GPIO_NUM     -1
61 #define RESET_GPIO_NUM    15
62 #define XCLK_GPIO_NUM     27
63 #define SIOD_GPIO_NUM     22
64 #define SIOC_GPIO_NUM     23
65
66 #define Y9_GPIO_NUM       19
67 #define Y8_GPIO_NUM       36
68 #define Y7_GPIO_NUM       18
69 #define Y6_GPIO_NUM       39
70 #define Y5_GPIO_NUM        5
71 #define Y4_GPIO_NUM       34
72 #define Y3_GPIO_NUM       35
73 #define Y2_GPIO_NUM       32
74 #define VSYNC_GPIO_NUM    25
75 #define HREF_GPIO_NUM     26
76 #define PCLK_GPIO_NUM     21
77
78 #elif defined(CAMERA_MODEL_M5STACK_WIDE)
79 #define PWDN_GPIO_NUM     -1
80 #define RESET_GPIO_NUM    15
81 #define XCLK_GPIO_NUM     27
82 #define SIOD_GPIO_NUM     22
83 #define SIOC_GPIO_NUM     23
84
85 #define Y9_GPIO_NUM       19
86 #define Y8_GPIO_NUM       36
87 #define Y7_GPIO_NUM       18
88 #define Y6_GPIO_NUM       39
89 #define Y5_GPIO_NUM        5
90 #define Y4_GPIO_NUM       34
91 #define Y3_GPIO_NUM       35
92 #define Y2_GPIO_NUM       32
93 #define VSYNC_GPIO_NUM    25
94 #define HREF_GPIO_NUM     26
95 #define PCLK_GPIO_NUM     21
96
97 #elif defined(CAMERA_MODEL_M5STACK_ESP32CAM)
98 #define PWDN_GPIO_NUM     -1
99 #define RESET_GPIO_NUM    15
100 #define XCLK_GPIO_NUM     27
101 #define SIOD_GPIO_NUM     25
102 #define SIOC_GPIO_NUM     23
103
104 #define Y9_GPIO_NUM       19
105 #define Y8_GPIO_NUM       36
106 #define Y7_GPIO_NUM       18
107 #define Y6_GPIO_NUM       39
108 #define Y5_GPIO_NUM        5
109 #define Y4_GPIO_NUM       34
110 #define Y3_GPIO_NUM       35
111 #define Y2_GPIO_NUM       17
112 #define VSYNC_GPIO_NUM    22
113 #define HREF_GPIO_NUM     26
114 #define PCLK_GPIO_NUM     21
115
116 #elif defined(CAMERA_MODEL_AI_THINKER)
117 #define PWDN_GPIO_NUM     32
118 #define RESET_GPIO_NUM    -1
119 #define XCLK_GPIO_NUM      0
120 #define SIOD_GPIO_NUM     26
121 #define SIOC_GPIO_NUM     27
122
123 #define Y9_GPIO_NUM       35
124 #define Y8_GPIO_NUM       34
125 #define Y7_GPIO_NUM       39
126 #define Y6_GPIO_NUM       36
127 #define Y5_GPIO_NUM       21
128 #define Y4_GPIO_NUM       19
129 #define Y3_GPIO_NUM       18
130 #define Y2_GPIO_NUM        5
131 #define VSYNC_GPIO_NUM    25
132 #define HREF_GPIO_NUM     23
133 #define PCLK_GPIO_NUM     22
134
135 #elif defined(CAMERA_MODEL_TTGO_T_JOURNAL)
136 #define PWDN_GPIO_NUM      0
137 #define RESET_GPIO_NUM    15
138 #define XCLK_GPIO_NUM     27
```

```
139 #define SIOD_GPIO_NUM     25
140 #define SIOC_GPIO_NUM     23
141
142 #define Y9_GPIO_NUM       19
143 #define Y8_GPIO_NUM       36
144 #define Y7_GPIO_NUM       18
145 #define Y6_GPIO_NUM       39
146 #define Y5_GPIO_NUM        5
147 #define Y4_GPIO_NUM       34
148 #define Y3_GPIO_NUM       35
149 #define Y2_GPIO_NUM       17
150 #define VSYNC_GPIO_NUM    22
151 #define HREF_GPIO_NUM     26
152 #define PCLK_GPIO_NUM     21
153
154 #else
155 #error "Camera model not selected"
156 #endif
```

5.29 WiFiTelemetry/camSD.cpp File Reference

SD card routines.

```
#include "camSD.h"
```

Functions

- void listDir (fs::FS &fs, const char ∗dirname, uint8_t levels)

  List dir in SD card.
- void createDir (fs::FS &fs, const char ∗path)

  Create a Dir object.
- void writeFile (fs::FS &fs, const char ∗path, const char ∗message)

  Write a file in SD card.
- void readConfigFile (fs::FS &fs)

  Read the config flight file.
- void updateConfigFile (fs::FS &fs)

  Updates the 'src\config.txt' file in the SD card.
- void writeDataLogFlight (fs::FS &fs)

  Writes on the SD card the flight data.
- void setupSD ()

  Routine called in the setup().

Variables

- const char ∗ configFilePath = "/src/config.txt"
- const char ∗ flightDataPath = "/data/flightData"
- const char ∗ logFileName
- int numberOfDataFiles = 0
- uint8_t isConnectedSD = 0

5.29.1 Detailed Description

SD card routines.

Author

@sebastiano123-c

See https://gist.github.com/youjunjer/b70b6e54ae7201a46387b8e73894ba51 for further details.

Version

0.1

Date

2022-03-01

Copyright

Copyright (c) 2022

5.29.2 Function Documentation

5.29.2.1 createDir()

```
void createDir (
            fs::FS & fs,
            const char * path )
```

Create a Dir object.

Parameters

fs	
path	

5.29.2.2 listDir()

```
void listDir (
            fs::FS & fs,
            const char * dirname,
            uint8_t levels )
```

List dir in SD card.

Parameters

fs	
dirname	
levels	

5.29.2.3 readConfigFile()

```
void readConfigFile (
            fs::FS & fs )
```

Read the config flight file.

5.29.2.4 setupSD()

```
void setupSD ( )
```

Routine called in the setup().

This setup routine checks if the SD is attached. Then creates, if not yet done, the data folder and instantiates the flight data file header. Finally, sends to DroneIno the initial PID parameters found on the config.txt file.

5.29.2.5 updateConfigFile()

```
void updateConfigFile (
            fs::FS & fs )
```

Updates the 'src\config.txt' file in the SD card.

Parameters

fs	

5.29.2.6 writeDataLogFlight()

```
void writeDataLogFlight (
            fs::FS & fs )
```

Writes on the SD card the flight data.

Parameters

fs	

5.29.2.7 writeFile()

```
void writeFile (
            fs::FS & fs,
            const char * path,
            const char * message )
```

Write a file in SD card.

Parameters

fs	
path	
message	

5.29.3 Variable Documentation

5.29.3.1 configFilePath

```
const char* configFilePath = "/src/config.txt"
```

5.29.3.2 flightDataPath

```
const char* flightDataPath = "/data/flightData"
```

5.29.3.3 isConnectedSD

```
uint8_t isConnectedSD = 0
```

5.29.3.4 logFileName

```
const char* logFileName
```

5.29.3.5 numberOfDataFiles

```
int numberOfDataFiles = 0
```

5.30 WiFiTelemetry/camSD.h File Reference

```
#include "FS.h"
#include "SD_MMC.h"
#include "telemetry.h"
#include "time.h"
```

Functions

- void setupSD ()

  Routine called in the setup().
- void writeDataLogFlight (fs::FS &fs)

  Writes on the SD card the flight data.
- void updateConfigFile (fs::FS &fs)

  Updates the 'src\config.txt' file in the SD card.

Variables

- uint8_t isConnectedSD

5.30.1 Function Documentation

5.30.1.1 setupSD()

```
void setupSD ( )
```

Routine called in the setup().

This setup routine checks if the SD is attached. Then creates, if not yet done, the data folder and instantiates the flight data file header. Finally, sends to DroneIno the initial PID parameters found on the config.txt file.

5.30.1.2 updateConfigFile()

```
void updateConfigFile (
            fs::FS & fs )
```

Updates the 'src\config.txt' file in the SD card.

Parameters

fs	

5.30.1.3 writeDataLogFlight()

```
void writeDataLogFlight (
            fs::FS & fs )
```

Writes on the SD card the flight data.

Parameters

fs	

5.30.2 Variable Documentation

5.30.2.1 isConnectedSD

```
uint8_t isConnectedSD  [extern]
```

5.31 camSD.h

Go to the documentation of this file.

```
1 #pragma once
2
3 #include "FS.h"
4 #include "SD_MMC.h"
5 #include "telemetry.h"
6 #include "time.h"
7
8 extern uint8_t isConnectedSD;
9
10 extern void setupSD();
11 extern void writeDataLogFlight(fs::FS &fs);
12 extern void updateConfigFile(fs::FS &fs);
```

5.32 WiFiTelemetry/telemetry.cpp File Reference

Telemtry and PID routines.

```
#include "telemetry.h"
```

Functions

- HardwareSerial SUART (1)
- void beginUARTCOM ()
- void updatePID ()
- void writeDataTransfer ()
- void readDataTransfer ()

Variables

- float dataController [dataControllerSize]
- float dataTransfer [dataTransferSize]

5.32.1 Detailed Description

Telemtry and PID routines.

Author

@sebastiano123-c

Version

0.1

Date

2022-03-01

Copyright

5.32.2 Function Documentation

5.32.2.1 beginUARTCOM()

```
void beginUARTCOM ( )
```

5.32.2.2 readDataTransfer()

```
void readDataTransfer ( )
```

5.32.2.3 SUART()

```
HardwareSerial SUART (
             1  )
```

5.32.2.4 updatePID()

```
void updatePID ( )
```

5.32.2.5 writeDataTransfer()

```
void writeDataTransfer ( )
```

5.32.3 Variable Documentation

5.32.3.1 dataController

```
float dataController[dataControllerSize]
```

5.32.3.2 dataTransfer

```
float dataTransfer[dataTransferSize]
```

5.33 WiFiTelemetry/telemetry.h File Reference

```
#include "app_httpd.h"
#include "HardwareSerial.h"
```

Functions

- void writeDataTransfer ()
- void readDataTransfer ()
- void beginUARTCOM ()

Variables

- float PID_P_GAIN_ROLL

  PID parameters for roll, pitch, yaw and altitude.
- float PID_I_GAIN_ROLL
- float PID_D_GAIN_ROLL
- float PID_P_GAIN_PITCH
- float PID_I_GAIN_PITCH
- float PID_D_GAIN_PITCH
- float PID_P_GAIN_YAW
- float PID_I_GAIN_YAW
- float PID_D_GAIN_YAW
- float GYROSCOPE_ROLL_FILTER

  GYROSCOPE.
- float GYROSCOPE_ROLL_CORR
- float GYROSCOPE_PITCH_CORR
- float PID_P_GAIN_ALTITUDE
- float PID_I_GAIN_ALTITUDE
- float PID_D_GAIN_ALTITUDE
- float rollAngle
- float pitchAngle
- float flightMode
- float batteryPercentage
- float altitudeMeasure
- const int dataTransferSize = 5
- const int dataControllerSize = 12

5.33.1 Function Documentation

5.33.1.1 beginUARTCOM()

```
void beginUARTCOM ( )
```

5.33.1.2 readDataTransfer()

```
void readDataTransfer ( )
```

5.33.1.3 writeDataTransfer()

```
void writeDataTransfer ( )
```

5.33.2 Variable Documentation

5.33.2.1 altitudeMeasure

```
float altitudeMeasure  [extern]
```

5.33.2.2 batteryPercentage

```
float batteryPercentage  [extern]
```

5.33.2.3 dataControllerSize

```
const int dataControllerSize = 12
```

5.33.2.4 dataTransferSize

```
const int dataTransferSize = 5
```

5.33.2.5 flightMode

```
float flightMode  [extern]
```

5.33.2.6 GYROSCOPE_PITCH_CORR

```
float GYROSCOPE_PITCH_CORR  [extern]
```

5.33.2.7 GYROSCOPE_ROLL_CORR

```
float GYROSCOPE_ROLL_CORR  [extern]
```

5.33.2.8 GYROSCOPE_ROLL_FILTER

```
float GYROSCOPE_ROLL_FILTER [extern]
```

GYROSCOPE.

5.33.2.9 PID_D_GAIN_ALTITUDE

```
float PID_D_GAIN_ALTITUDE [extern]
```

5.33.2.10 PID_D_GAIN_PITCH

```
float PID_D_GAIN_PITCH [extern]
```

5.33.2.11 PID_D_GAIN_ROLL

```
float PID_D_GAIN_ROLL [extern]
```

5.33.2.12 PID_D_GAIN_YAW

```
float PID_D_GAIN_YAW [extern]
```

5.33.2.13 PID_I_GAIN_ALTITUDE

```
float PID_I_GAIN_ALTITUDE [extern]
```

5.33.2.14 PID_I_GAIN_PITCH

```
float PID_I_GAIN_PITCH [extern]
```

5.33.2.15 PID_I_GAIN_ROLL

```
float PID_I_GAIN_ROLL  [extern]
```

5.33.2.16 PID_I_GAIN_YAW

```
float PID_I_GAIN_YAW  [extern]
```

5.33.2.17 PID_P_GAIN_ALTITUDE

```
float PID_P_GAIN_ALTITUDE  [extern]
```

5.33.2.18 PID_P_GAIN_PITCH

```
float PID_P_GAIN_PITCH  [extern]
```

5.33.2.19 PID_P_GAIN_ROLL

```
float PID_P_GAIN_ROLL  [extern]
```

PID parameters for roll, pitch, yaw and altitude.

Note

Roll and pitch parameters have the same values.

5.33.2.20 PID_P_GAIN_YAW

```
float PID_P_GAIN_YAW  [extern]
```

5.33.2.21 pitchAngle

```
float pitchAngle  [extern]
```

5.33.2.22 rollAngle

```
float rollAngle  [extern]
```

5.34 telemetry.h

Go to the documentation of this file.

```
1 #pragma once
2 #include "app_httpd.h"
3 #include "HardwareSerial.h"
4
5
6 //              (ROLL)
7 extern float PID_P_GAIN_ROLL;                    //Gain setting for the roll P-controller (1.3)
8 extern float PID_I_GAIN_ROLL;                   //Gain setting for the roll I-controller  (0.0002)
9 extern float PID_D_GAIN_ROLL;                    //Gain setting for the roll D-controller (10.0)
10
11 //              (PITCH)
12 extern float PID_P_GAIN_PITCH;          //Gain setting for the pitch P-controller
13 extern float PID_I_GAIN_PITCH;          //Gain setting for the pitch I-controller
14 extern float PID_D_GAIN_PITCH;          //Gain setting for the pitch D-controller
15
16 //              (YAW)
17 extern float PID_P_GAIN_YAW;                      //Gain setting for the pitch P-controller. (2.0)
18 extern float PID_I_GAIN_YAW;                     //Gain setting for the pitch I-controller. (0.04)
19 extern float PID_D_GAIN_YAW;                      //Gain setting for the pitch D-controller. (0.0)
20
21 // GYROSCOPE
22 extern float GYROSCOPE_ROLL_FILTER;                       // read your gyroscope data after the
       calibration, try different values and choose the best one
23 extern float GYROSCOPE_ROLL_CORR;                       // (0.) after set GYROSCOPE_ROLL_FILTER, put here
       the angle roll you read eneabling DEBUG
24 extern float GYROSCOPE_PITCH_CORR;                       // (-1.65.) after set GYROSCOPE_PITCH_FILTER, put
       here the angle pitch you read eneabling DEBUG
25
26 //              (ALTITUDE)

27 extern float PID_P_GAIN_ALTITUDE;                    //Gain setting for the pitch P-controller. (2.0)
28 extern float PID_I_GAIN_ALTITUDE;                    //Gain setting for the pitch I-controller. (0.04)
29 extern float PID_D_GAIN_ALTITUDE;                     //Gain setting for the pitch D-controller. (0.0)
30
31 extern float rollAngle;
32 extern float pitchAngle;
33 extern float flightMode;
34 extern float batteryPercentage;
35 extern float altitudeMeasure;
36
37 const int dataTransferSize = 5;
38 const int dataControllerSize = 12;
39
40 extern void writeDataTransfer();
41 extern void readDataTransfer();
42 extern void beginUARTCOM();
```

5.35 WiFiTelemetry/WiFiTelemetry.ino File Reference

```
#include <Arduino.h>
#include "esp_camera.h"
#include <WiFi.h>
#include "camera_pins.h"
#include "app_httpd.h"
#include "telemetry.h"
#include "camSD.h"
```

Macros

- #define CAMERA_MODEL_AI_THINKER

Functions

- void setup ()
- void loop ()

Variables

- const char * ssid = "DroneInoTelemetry"
- const char * password = "DroneIno"
- const int timeDelay = 1
- float PID_P_GAIN_ROLL = 0.

  PID parameters for roll, pitch, yaw and altitude.
- float PID_I_GAIN_ROLL = 0.
- float PID_D_GAIN_ROLL = 0.
- float PID_P_GAIN_PITCH = 0.
- float PID_I_GAIN_PITCH = 0.
- float PID_D_GAIN_PITCH = 0.
- float PID_P_GAIN_YAW = 0.
- float PID_I_GAIN_YAW = 0.
- float PID_D_GAIN_YAW = 0.
- float GYROSCOPE_ROLL_FILTER = 0.

  GYROSCOPE.
- float GYROSCOPE_ROLL_CORR = 0.
- float GYROSCOPE_PITCH_CORR = 0.
- float PID_P_GAIN_ALTITUDE = 0.
- float PID_I_GAIN_ALTITUDE = 0.
- float PID_D_GAIN_ALTITUDE = 0.
- float rollAngle = 1.
- float pitchAngle = 1.
- float flightMode = 1.
- float batteryPercentage = 1.
- float altitudeMeasure = 1.

5.35.1 Macro Definition Documentation

5.35.1.1 CAMERA_MODEL_AI_THINKER

```
#define CAMERA_MODEL_AI_THINKER
```

5.35.2 Function Documentation

5.35.2.1 loop()

```
void loop ( )
```

5.35.2.2 setup()

```
void setup ( )
```

5.35.3 Variable Documentation

5.35.3.1 altitudeMeasure

```
float altitudeMeasure = 1.
```

5.35.3.2 batteryPercentage

```
float batteryPercentage = 1.
```

5.35.3.3 flightMode

```
float flightMode = 1.
```

5.35.3.4 GYROSCOPE_PITCH_CORR

```
float GYROSCOPE_PITCH_CORR = 0.
```

5.35.3.5 GYROSCOPE_ROLL_CORR

```
float GYROSCOPE_ROLL_CORR = 0.
```

5.35.3.6 GYROSCOPE_ROLL_FILTER

```
float GYROSCOPE_ROLL_FILTER = 0.
```

GYROSCOPE.

5.35.3.7 password

```
const char* password = "DroneIno"
```

5.35.3.8 PID_D_GAIN_ALTITUDE

```
float PID_D_GAIN_ALTITUDE = 0.
```

5.35.3.9 PID_D_GAIN_PITCH

```
float PID_D_GAIN_PITCH = 0.
```

5.35.3.10 PID_D_GAIN_ROLL

```
float PID_D_GAIN_ROLL = 0.
```

5.35.3.11 PID_D_GAIN_YAW

```
float PID_D_GAIN_YAW = 0.
```

5.35.3.12 PID_I_GAIN_ALTITUDE

```
float PID_I_GAIN_ALTITUDE = 0.
```

5.35.3.13 PID_I_GAIN_PITCH

```
float PID_I_GAIN_PITCH = 0.
```

5.35.3.14 PID_I_GAIN_ROLL

```
float PID_I_GAIN_ROLL = 0.
```

5.35.3.15 PID_I_GAIN_YAW

```
float PID_I_GAIN_YAW = 0.
```

5.35.3.16 PID_P_GAIN_ALTITUDE

```
float PID_P_GAIN_ALTITUDE = 0.
```

5.35.3.17 PID_P_GAIN_PITCH

```
float PID_P_GAIN_PITCH = 0.
```

5.35.3.18 PID_P_GAIN_ROLL

```
float PID_P_GAIN_ROLL = 0.
```

PID parameters for roll, pitch, yaw and altitude.

Note

Roll and pitch parameters have the same values.

5.35.3.19 PID_P_GAIN_YAW

```
float PID_P_GAIN_YAW = 0.
```

5.35.3.20 pitchAngle

```
float pitchAngle = 1.
```

5.35.3.21 rollAngle

```
float rollAngle = 1.
```

5.35.3.22 ssid

```
const char* ssid = "DroneInoTelemetry"
```

5.35.3.23 timeDelay

```
const int timeDelay = 1
```

Index