/\*

Thread.h - An runnable object

Thread is responsable for holding the "action" for something,

also, it responds if it "should" or "should not" run, based on

the current time;

For instructions, go to https://github.com/ivanseidel/ArduinoThread

Created by Ivan Seidel Gomes, March, 2013.

Released into the public domain.

\*/

#ifndef Thread\_h

#define Thread\_h

#if defined(ARDUINO) && ARDUINO >= 100

#include <Arduino.h>

#else

#include <WProgram.h>

#endif

#include <inttypes.h>

/\*

Uncomment this line to enable ThreadName Strings.

It might be usefull if you are loging thread with Serial,

or displaying a list of threads...

\*/

// #define USE\_THREAD\_NAMES 1

class Thread{

protected:

// Desired interval between runs

unsigned long interval;

// Last runned time in Ms

unsigned long last\_run;

// Scheduled run in Ms (MUST BE CACHED)

unsigned long \_cached\_next\_run;

/\*

IMPORTANT! Run after all calls to run()

Updates last\_run and cache next run.

NOTE: This MUST be called if extending

this class and implementing run() method

\*/

void runned(unsigned long time);

// Default is to mark it runned "now"

void runned() { runned(millis()); }

// Callback for run() if not implemented

void (\*\_onRun)(void);

public:

// If the current Thread is enabled or not

bool enabled;

// ID of the Thread (initialized from memory adr.)

int ThreadID;

#ifdef USE\_THREAD\_NAMES

// Thread Name (used for better UI).

String ThreadName;

#endif

Thread(void (\*callback)(void) = NULL, unsigned long \_interval = 0);

// Set the desired interval for calls, and update \_cached\_next\_run

virtual void setInterval(unsigned long \_interval);

// Return if the Thread should be runned or not

virtual bool shouldRun(unsigned long time);

// Default is to check whether it should run "now"

bool shouldRun() { return shouldRun(millis()); }

// Callback set

void onRun(void (\*callback)(void));

// Runs Thread

virtual void run();

};

#endif