Dobre praktyki, których nie ma na embedded

Panicz Maciej Godek

godek.maciek@gmail.com

Gdańsk Embedded Meetup#5, 04.02.2020

Bój się Boga

Bój się Boga
którego nie ma

Bój się Boga
którego nie ma
w Twoim sercu

Bój się Boga

~~kurwy nie~~

w Twoim sercu

Dobre praktyki których nie ma na embedded

Dobre praktyki

~~[illegible]~~

na embedded

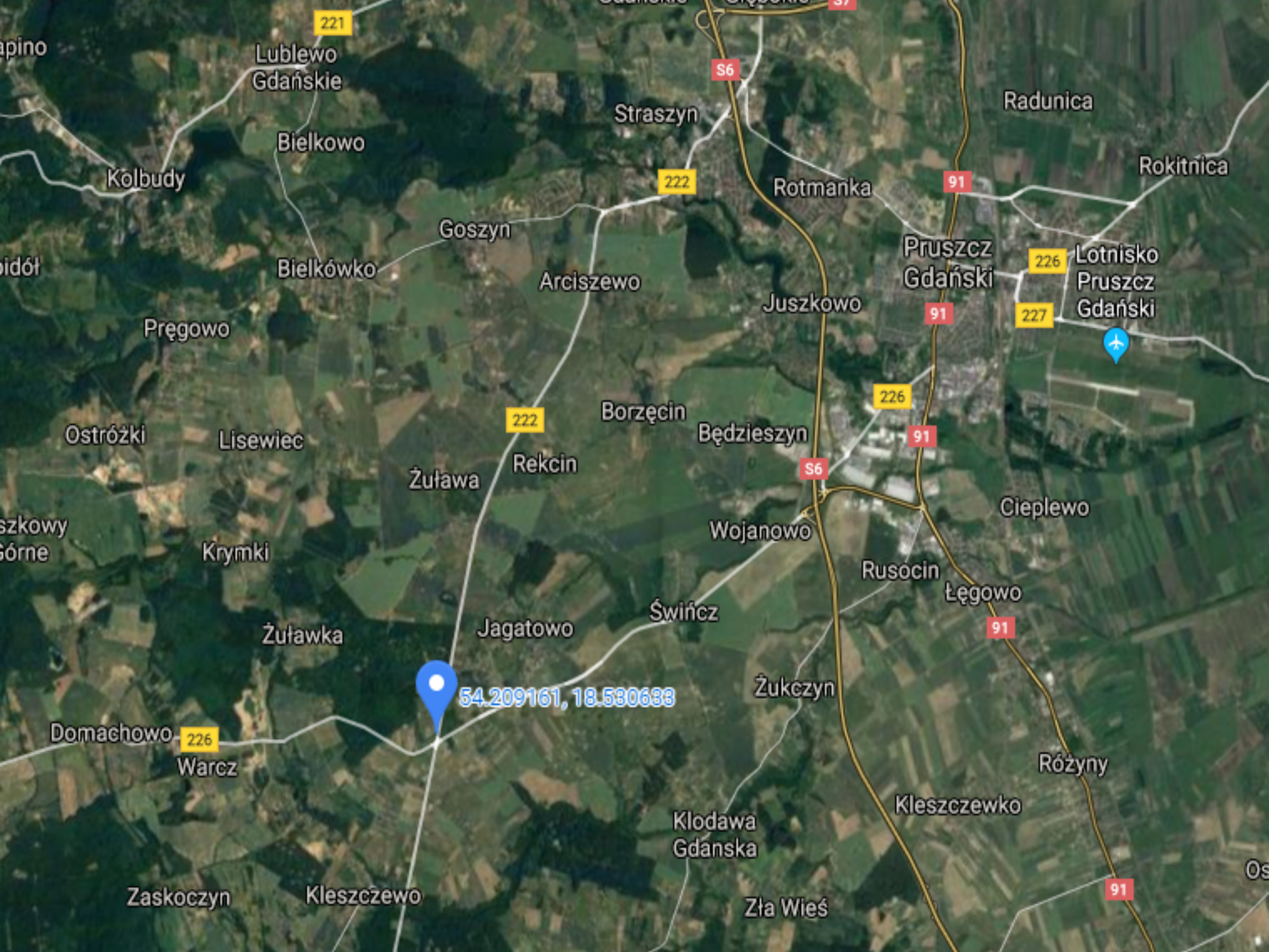

1
2
3
4
5
6
SUMA 0


1
2
3
4
5
6
SUMA 0


1
2
3
4
5
6
SUMA 0


1
2
3
4
5
6
SUMA 0


1
2
3
4
5
6
SUMA 0


221
Lublewo Gdańskie
S6
Straszyn
Radunica
Bielkowo
Kolbudy
222
Rotmanka
91
Rokitnica
Goszyn
Pruszcz Gdański
226
Lotnisko Pruszcz Gdański
Bielkówko
Arciszewo
Juszkowo
227
Pręgowo
Borzęcin
Ostróżki
Lisewiec
Będzieszyn
Rekcin
Żuława
Cieplewo
Wojanowo
Krymki
Rusocin
Łęgowo
Świńcz
Jagatowo
Żuławka
54.209161, 18.530633
Żukczyn
Domachowo
Warcz
Różyny
Kleszczewko
Klodawa Gdanska
Zaskoczyn
Kleszczewo
Zła Wieś

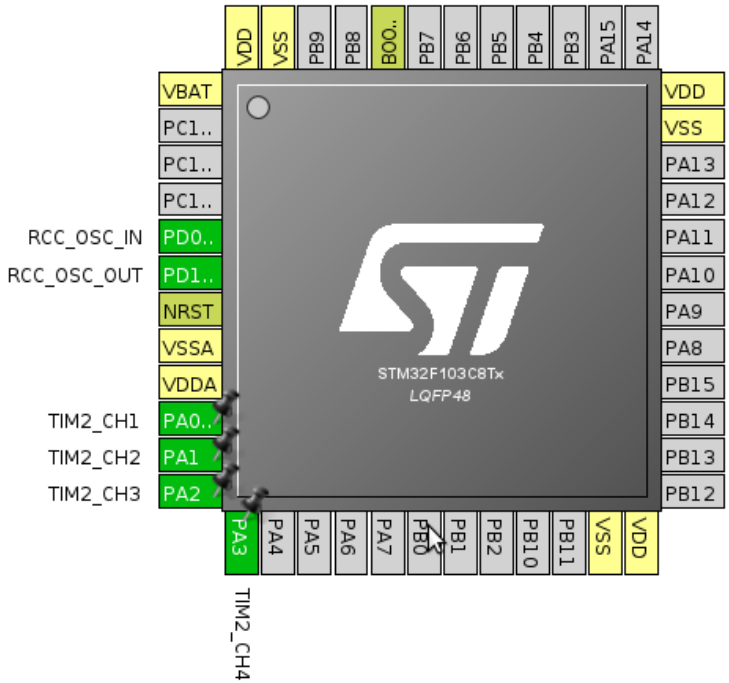
STM32CubeMX PWM_like_a_boss.ioc: STM32F103C8Tx

File Project Pinout Window Help

Keep Current Signals Placement | Find | Show user Label

Pinout | Clock Configuration | Configuration | Power Consumption Calculator

- CRC
- I2C1
- I2C2
- IWDG
- RCC
  - High Speed Clock (HSE): Crystal/Ceramic Resona...
  - Low Speed Clock (LSE): Disable
  - Master Clock Output
- RTC
- SPI1
- SPI2
- SYS
- TIM1
- TIM2
  - Slave Mode: Disable
  - Trigger Source: Disable
  - Clock Source: Internal Clock
  - Channel1: PWM Generation CH1
  - Channel2: PWM Generation CH2
  - Channel3: PWM Generation CH3
  - Channel4: PWM Generation CH4
  - Combined Channels: Disable
  - Use ETR as Clearing Source
  - XOR activation
  - One Pulse Mode
- TIM3
- TIM4
- USART1
- USART2
- USART3
- USB
- WWDG



MCUs Selection | Output

Series	Lines	Mcu	Package	Required Peripherals
STM32F1	STM32F103	STM32F103C4Tx	LQFP48	None
STM32F1	STM32F103	STM32F103C6Tx	LQFP48	None
STM32F1	STM32F103	STM32F103C8Tx	LQFP48	None

Przykład złego interfejsu

```
HAL_StatusTypeDef HAL_RTC_GetTime(RTC_HandleTypeDef *hrtc,
                                  RTC_TimeTypeDef *sTime,
                                  uint32_t Format)
```

Przykład złego interfejsu

```
HAL_StatusTypeDef HAL_RTC_GetTime(RTC_HandleTypeDef *hrtc,
                                  RTC_TimeTypeDef *sTime,
                                  uint32_t Format)

typedef struct {
    uint8_t Hours;
    uint8_t Minutes;
    uint8_t Seconds;
    uint8_t TimeFormat;
    uint32_t SubSeconds;
    uint32_t SecondFraction;
    uint32_t DayLightSaving;
    uint32_t StoreOperation;
} RTC_TimeTypeDef;
```

Przykład złego interfejsu

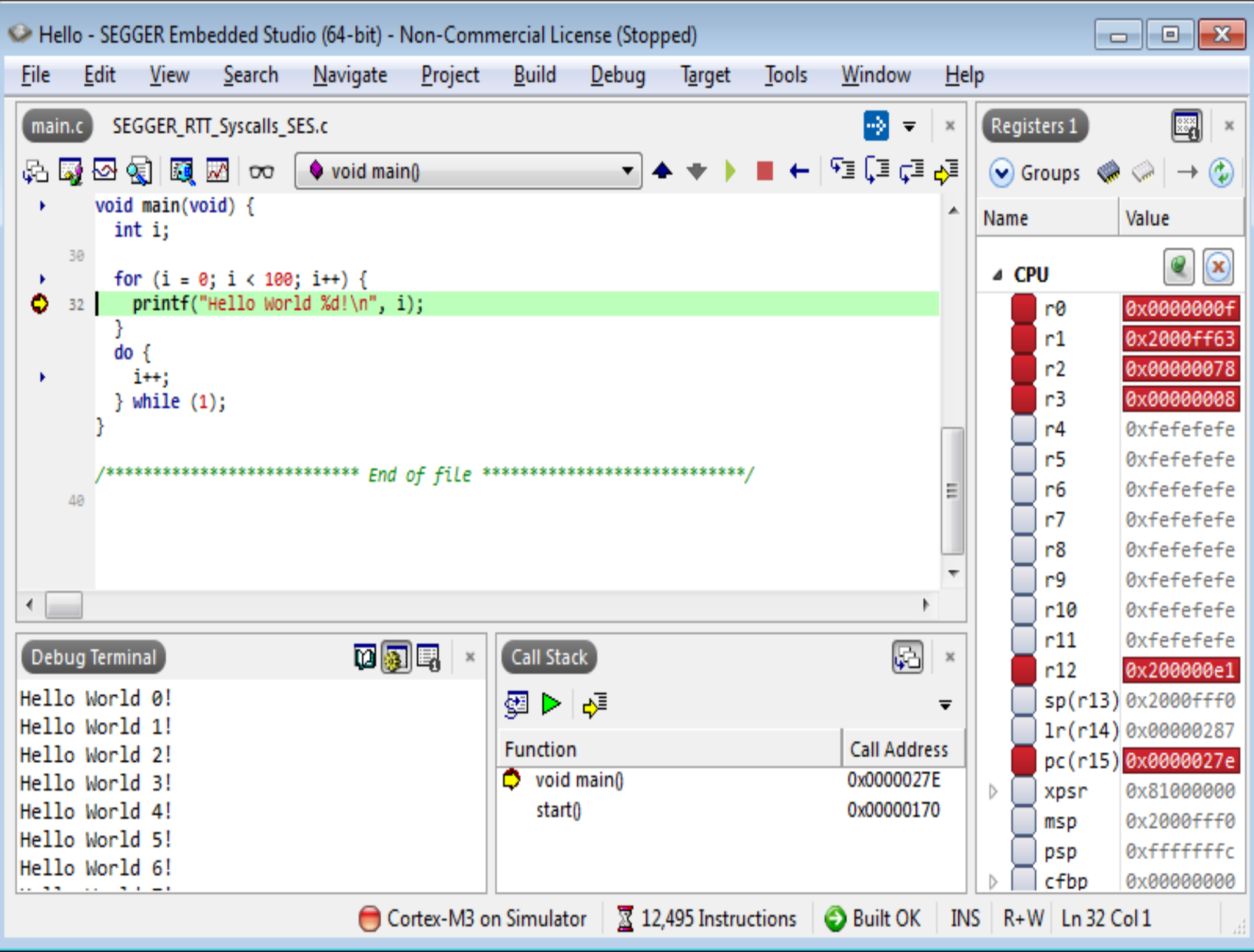
```
HAL_StatusTypeDef HAL_RTC_GetTime(RTC_HandleTypeDef *hrtc,
                                  RTC_TimeTypeDef *sTime,
                                  uint32_t Format)

typedef struct {
    uint8_t Hours;
    uint8_t Minutes;
    uint8_t Seconds;
    uint8_t TimeFormat;
    uint32_t SubSeconds;
    uint32_t SecondFraction;
    uint32_t DayLightSaving;
    uint32_t StoreOperation;
} RTC_TimeTypeDef;

/* @param  Format: Specifies the format of the
 *         entered parameters. This parameter can
 *         be one of the following values:
 *            @arg RTC_FORMAT_BIN: Binary data format
 *            @arg RTC_FORMAT_BCD: BCD data format */
```

Przykład złego interfejsu

```
/* @note You must call HAL_RTC_GetDate()
 *       after HAL_RTC_GetTime() to unlock
 *       the values in the higher-order
 *       calendar shadow registers to ensure
 *       consistency between the time
 *       and date values.
 *       Reading RTC current time locks
 *       the values in calendar shadow registers
 *       until current date is read. */
```

Inny przykład złego interfejsu

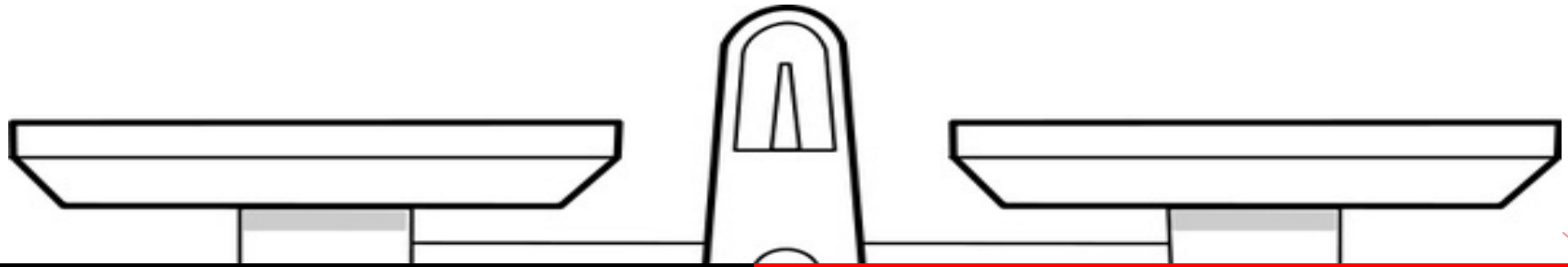
```
/**
 * @brief  Load the color lookup table.
 * @param  hltdc    pointer to a LTDC_HandleTypeDef
 *                  structure that contains the
 *                  configuration information for the LTDC.
 * @param  pCLUT    pointer to the color lookup table address.
 * @param  CLUTSize the color lookup table size.
 * @param  LayerIdx LTDC Layer index.
 *                  This parameter can be one
 *                  of the following values:
 *                  LTDC_LAYER_1 (0) or LTDC_LAYER_2 (1)
 * @retval HAL status
 */
HAL_StatusTypeDef
HAL_LTDC_ConfigCLUT(LTDC_HandleTypeDef *hltdc,
                    uint32_t *pCLUT,
                    uint32_t CLUTSize,
                    uint32_t LayerIdx)
```


Hello - SEGGER Embedded Studio (64-bit) - Non-Commercial License (Stopped)
File Edit View Search Navigate Project Build Debug Target Tools Window Help
main.c SEGGER_RTT_Syscalls_SES.c
void main()
void main(void) {
int i;
for (i = 0; i < 100; i++) {
printf("Hello World %d!\n", i);
}
do {
i++;
} while (1);
}
/*************************** End of file ****************************/
Registers 1
Groups
Name Value
CPU
r0 0x0000000f
r1 0x2000ff63
r2 0x00000078
r3 0x00000008
r4 0xfefefefe
r5 0xfefefefe
r6 0xfefefefe
r7 0xfefefefe
r8 0xfefefefe
r9 0xfefefefe
r10 0xfefefefe
r11 0xfefefefe
r12 0x200000e1
sp(r13) 0x2000fff0
lr(r14) 0x00000287
pc(r15) 0x0000027e
xpsr 0x81000000
msp 0x2000fff0
psp 0xfffffffc
cfbp 0x00000000
Debug Terminal
Hello World 0!
Hello World 1!
Hello World 2!
Hello World 3!
Hello World 4!
Hello World 5!
Hello World 6!
Call Stack
Function Call Address
void main() 0x0000027E
start() 0x00000170
Cortex-M3 on Simulator
12,495 Instructions
Built OK
INS
R+W
Ln 32 Col 1

Dedykowane IDE dla kontrolera

* klikamy DEBUG

  i się debuguje



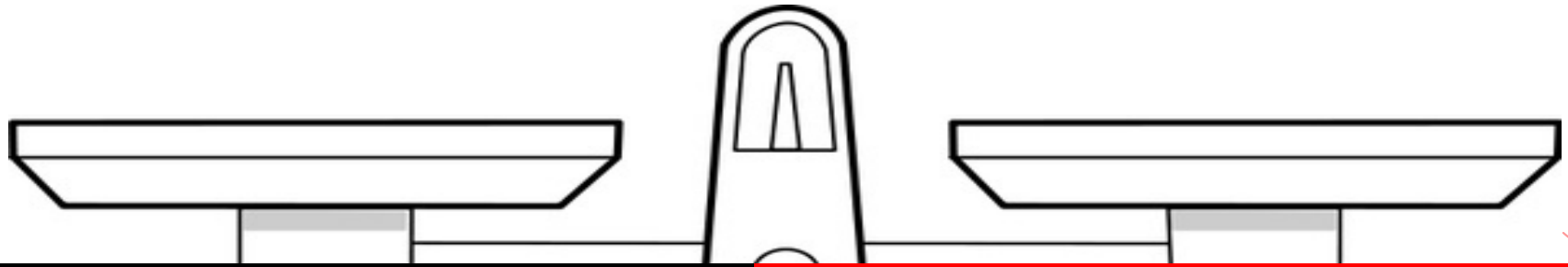
Dedykowane IDE dla kontrolera

* klikamy DEBUG

  i się debuguje

* za każdym razem musimy

  się uczyć od nowa



Dedykowane IDE dla kontrolera

* klikamy DEBUG

  i się debuguje

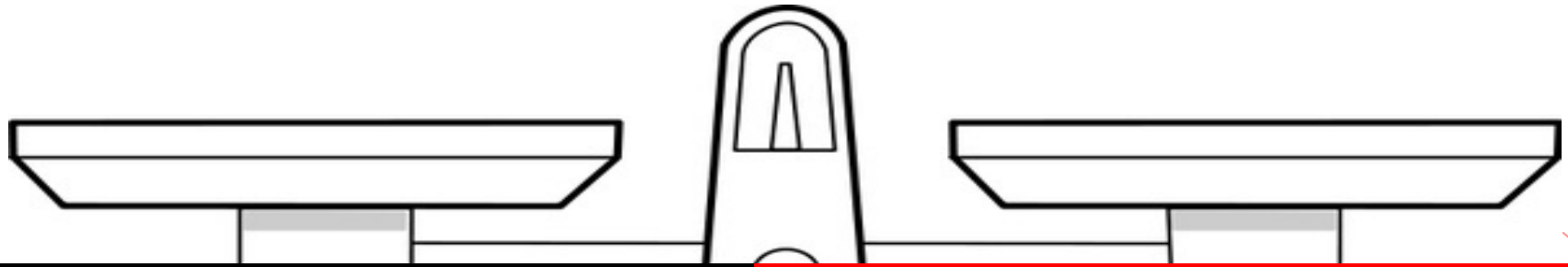
* robienie rzeczy, których

  nie przewidzieli twórcy,

  jest trudne (albo niemożliwe)

* za każdym razem musimy

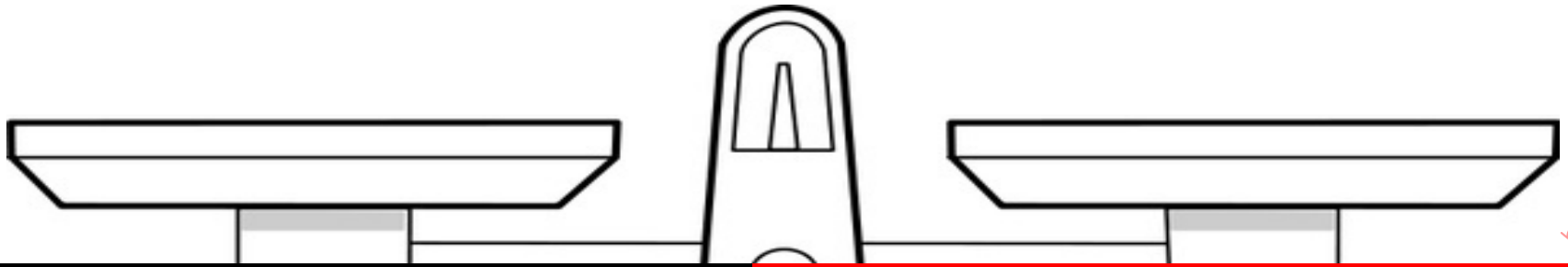
  się uczyć od nowa



Dedykowane IDE dla kontrolera

* to środowisko kontroluje nas,
  a nie my środowisko
* robienie rzeczy, których
  nie przewidzieli twórcy,
  jest trudne (albo niemożliwe)
* za każdym razem musimy
  się uczyć od nowa

* klikamy DEBUG
  i się debuguje



~

```
    Make sure you didn't emit any warnings while testing
msys/remake-git 4.1.2957.e3e34dd9-1
    Enhanced GNU Make - tracing, error reporting, debugging, profiling and more
msys/sharutils 4.15.2-1
    Makes so-called shell archives out of many files

Dorotka@LAPTOP-SR76FLAL MSYS ~
$ pacman -S make
rozwiązywanie zależności…
szukanie sprzecznych pakietów...

Pakiety (1) make-4.2.1-1

Do pobrania:        0,41 MiB
Do zainstalowania:  1,22 MiB

:: Kontynuować instalację? [T/n]
:: Pobieranie pakietów…
 make-4.2.1-1-x86_64                414,8 KiB   523K/s 00:01 [##################################] 100%
(1/1) sprawdzanie kluczy w bazie                              [##################################] 100%
(1/1) sprawdzanie spójności pakietów                          [##################################] 100%
(1/1) wczytywanie listy plików                                [##################################] 100%
(1/1) sprawdzanie konfliktów plików                           [##################################] 100%
(1/1) sprawdzanie dostępnego miejsca na dysku                 [##################################] 100%
:: Przetwarzanie zmian pakietu...
(1/1) instalowanie make                                       [##################################] 100%

Dorotka@LAPTOP-SR76FLAL MSYS ~
$
```

Zagadka - jak nazwać `f`?

```
(define (f lol)
  (apply map list lol))
```

Zagadka - jak nazwać `f`?

```
(define (f lol)
  (apply map list lol))

(list 1 2 3) ===> '(1 2 3)
```

Zagadka - jak nazwać `f`?

```
(define (f lol)
  (apply map list lol))

(list 1 2 3) ===> '(1 2 3)

(map + '(1 2 3) '(4 5 6))
   ===> (5 7 9)
```

Zagadka - jak nazwać `f`?

```
(define (f lol)
  (apply map list lol))

(list 1 2 3) ===> '(1 2 3)

(map + '(1 2 3) '(4 5 6))
   ===> (5 7 9)

(apply f '(1 2 3))
 === (f 1 2 3)
```

Zagadka - jak nazwać `f`?

```
(define (f lol)
  (apply map list lol))

(list 1 2 3) ===> '(1 2 3)

(map + '(1 2 3) '(4 5 6))
   ===> (5 7 9)

(apply f '(1 2 3))
 === (f 1 2 3)

(apply f 0 '(1 2 3))
 === (f 0 1 2 3)
```

Zagadka - jak nazwać `f`?

```
(define (f lol)
  (apply map list lol))
```

```
def f(lol):
    return map(list, *lol)
```

```
(list 1 2 3) ===> '(1 2 3)
```

```
(map + '(1 2 3) '(4 5 6))
   ===> (5 7 9)
```

```
(apply f '(1 2 3))
 === (f 1 2 3)
```

```
(apply f 0 '(1 2 3))
 === (f 0 1 2 3)
```

Zagadka - jak nazwać `f`?

```
(define (f lol)
  (apply map list lol))

(list 1 2 3) ===> '(1 2 3)

(map + '(1 2 3) '(4 5 6))
   ===> (5 7 9)

(apply f '(1 2 3))
 === (f 1 2 3)

(apply f 0 '(1 2 3))
 === (f 0 1 2 3)
```

```
def f(lol):
    return map(list, *lol)

def list(*elements):
    return elements
```

Zagadka - jak nazwać `f`?

```
(define (f lol)
  (apply map list lol))

(list 1 2 3) ===> '(1 2 3)

(map + '(1 2 3) '(4 5 6))
   ===> (5 7 9)

(apply f '(1 2 3))
 === (f 1 2 3)

(apply f 0 '(1 2 3))
 === (f 0 1 2 3)
```

```
def f(lol):
    return map(list, *lol)

def list(*elements):
    return elements

(apply map list '((1 2 3)
                  (4 5 6)))
```

Zagadka - jak nazwać `f`?

```
(define (f lol)
  (apply map list lol))

(list 1 2 3) ===> '(1 2 3)

(map + '(1 2 3) '(4 5 6))
   ===> (5 7 9)

(apply f '(1 2 3))
 === (f 1 2 3)

(apply f 0 '(1 2 3))
 === (f 0 1 2 3)
```

```
def f(lol):
    return map(list, *lol)

def list(*elements):
    return elements

(apply map list '((1 2 3)
                  (4 5 6)))

(map list '(1 2 3) '(4 5 6))
```

Zagadka - jak nazwać `f`?

```
(define (f lol)
  (apply map list lol))

(list 1 2 3) ===> '(1 2 3)

(map + '(1 2 3) '(4 5 6))
   ===> (5 7 9)

(apply f '(1 2 3))
 === (f 1 2 3)

(apply f 0 '(1 2 3))
 === (f 0 1 2 3)
```

```
def f(lol):
    return map(list, *lol)

def list(*elements):
    return elements

(apply map list '((1 2 3)
                  (4 5 6)))

(map list '(1 2 3) '(4 5 6))

'((1 4)
  (2 5)
  (3 6))
```

Zagadka - jak nazwać `f`?

```
(define (f lol)
  (apply map list lol))

(list 1 2 3) ===> '(1 2 3)

(map + '(1 2 3) '(4 5 6))
   ===> (5 7 9)

(apply f '(1 2 3))
 === (f 1 2 3)

(apply f 0 '(1 2 3))
 === (f 0 1 2 3)
```

```
def f(lol):
    return map(list, *lol)

def list(*elements):
    return elements

(apply map list '((1 2 3)
                  (4 5 6)))

(map list '(1 2 3) '(4 5 6))

'((1 4)
  (2 5)
  (3 6))

(f '((1 2 3)
     (4 5 6))) ===> '((1 4)
                      (2 5)
                      (3 6))
```

Dobre praktyki

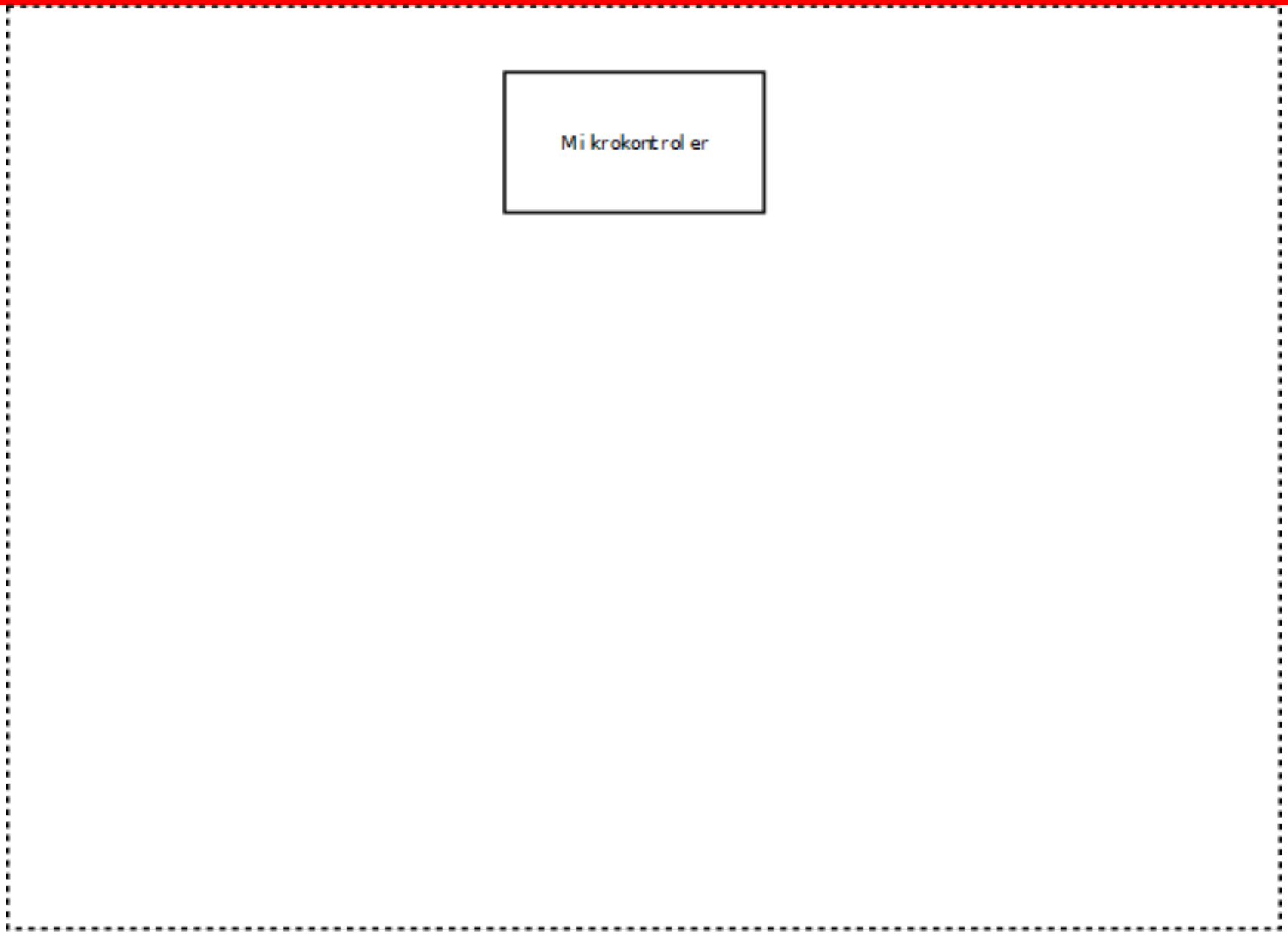
Dobre praktyki

- Testy jednostkowe

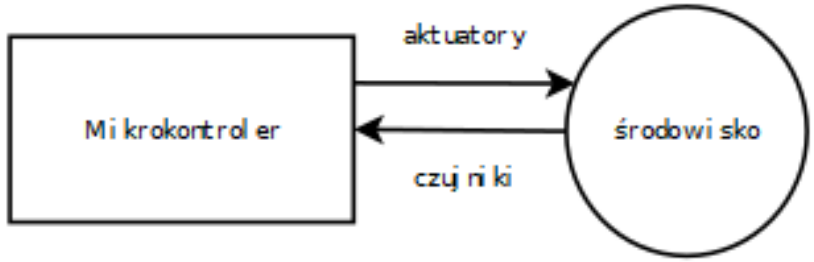
Dobre praktyki

- Testy jednostkowe
- Kod, który działa na kontrolerze, powinno dać się również wybudować i uruchomić na PC

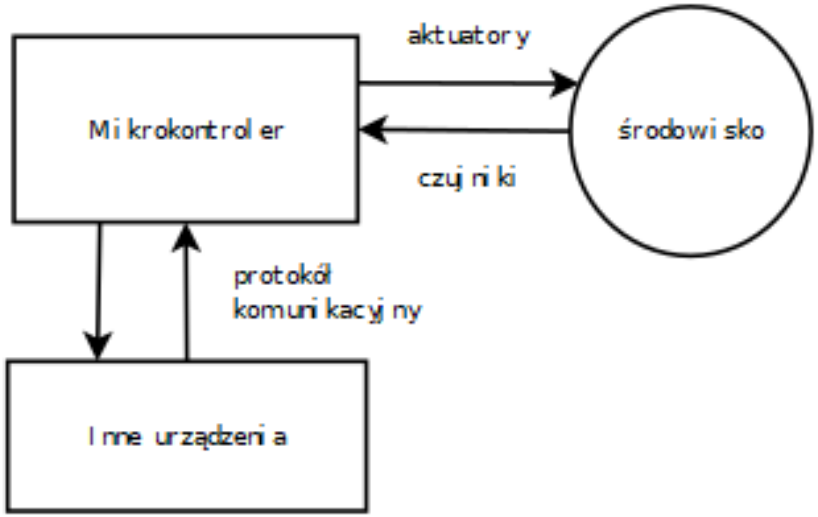
Struktura projektu embedded



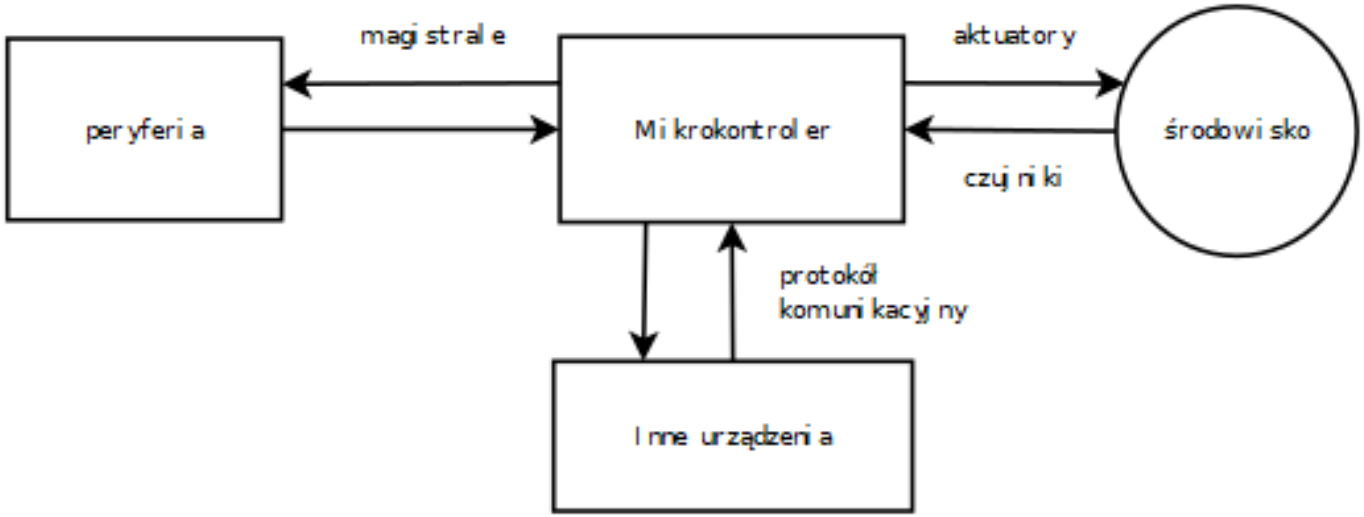
Struktura projektu embedded



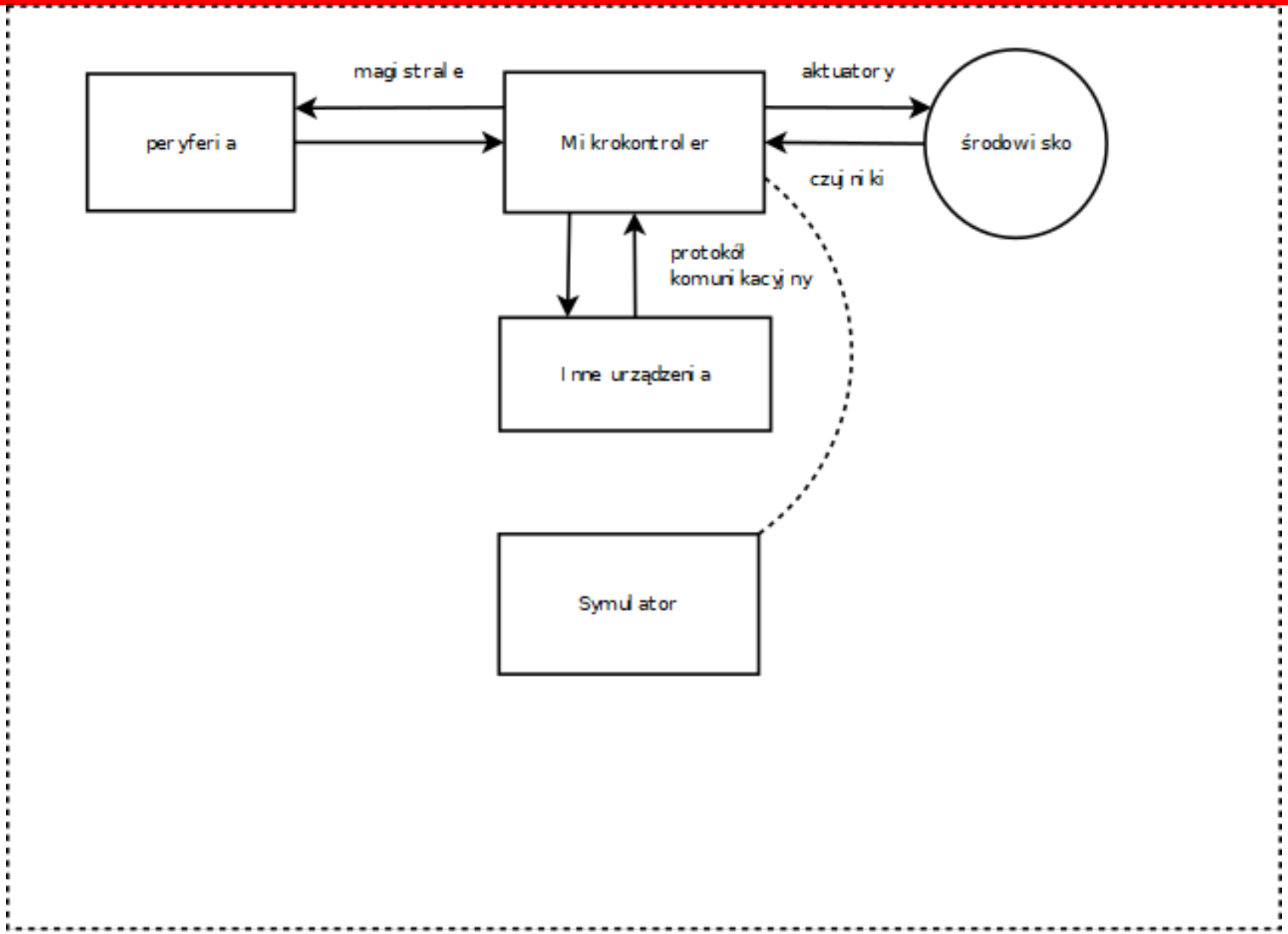
Struktura projektu embedded



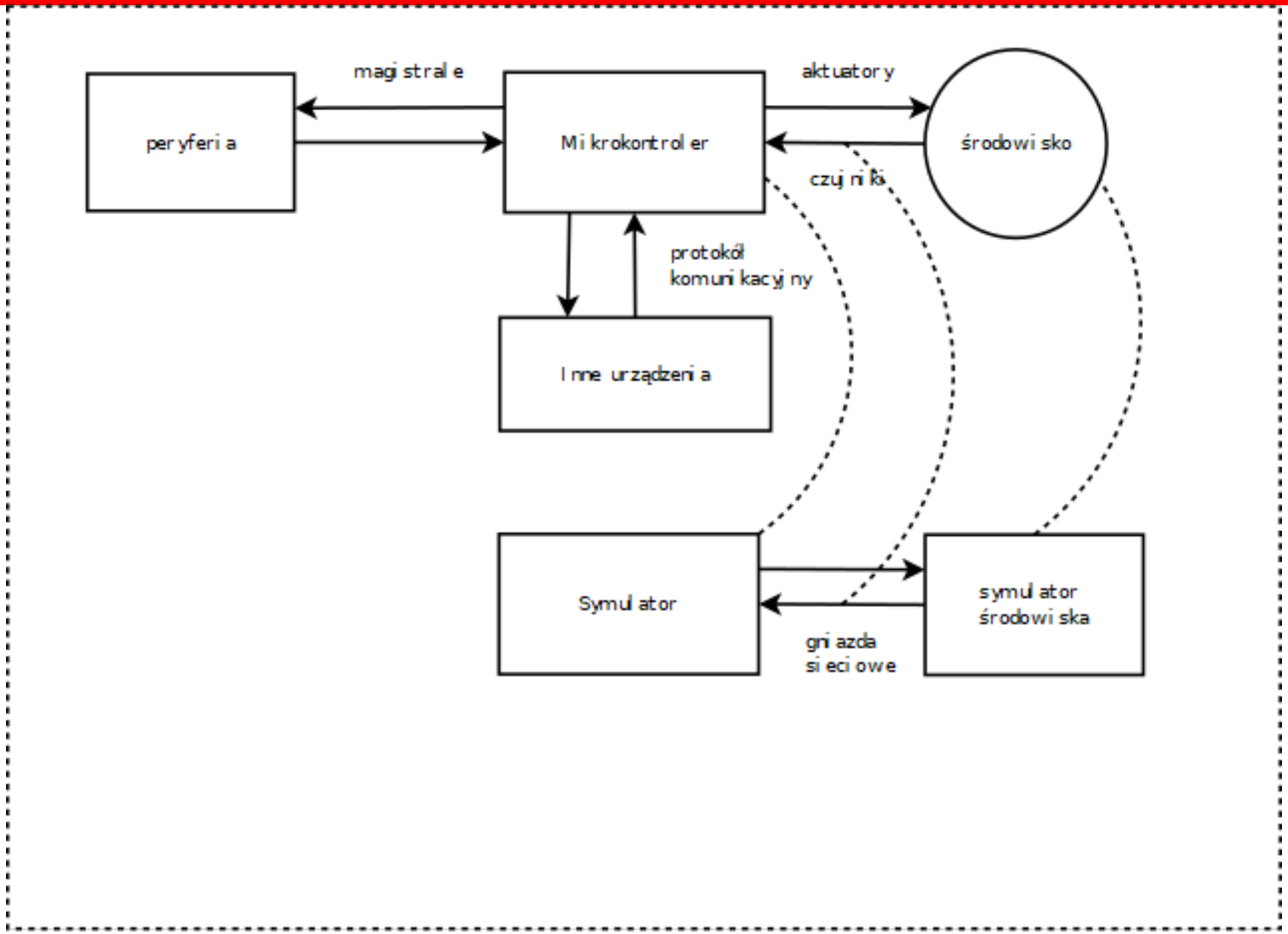
Struktura projektu embedded



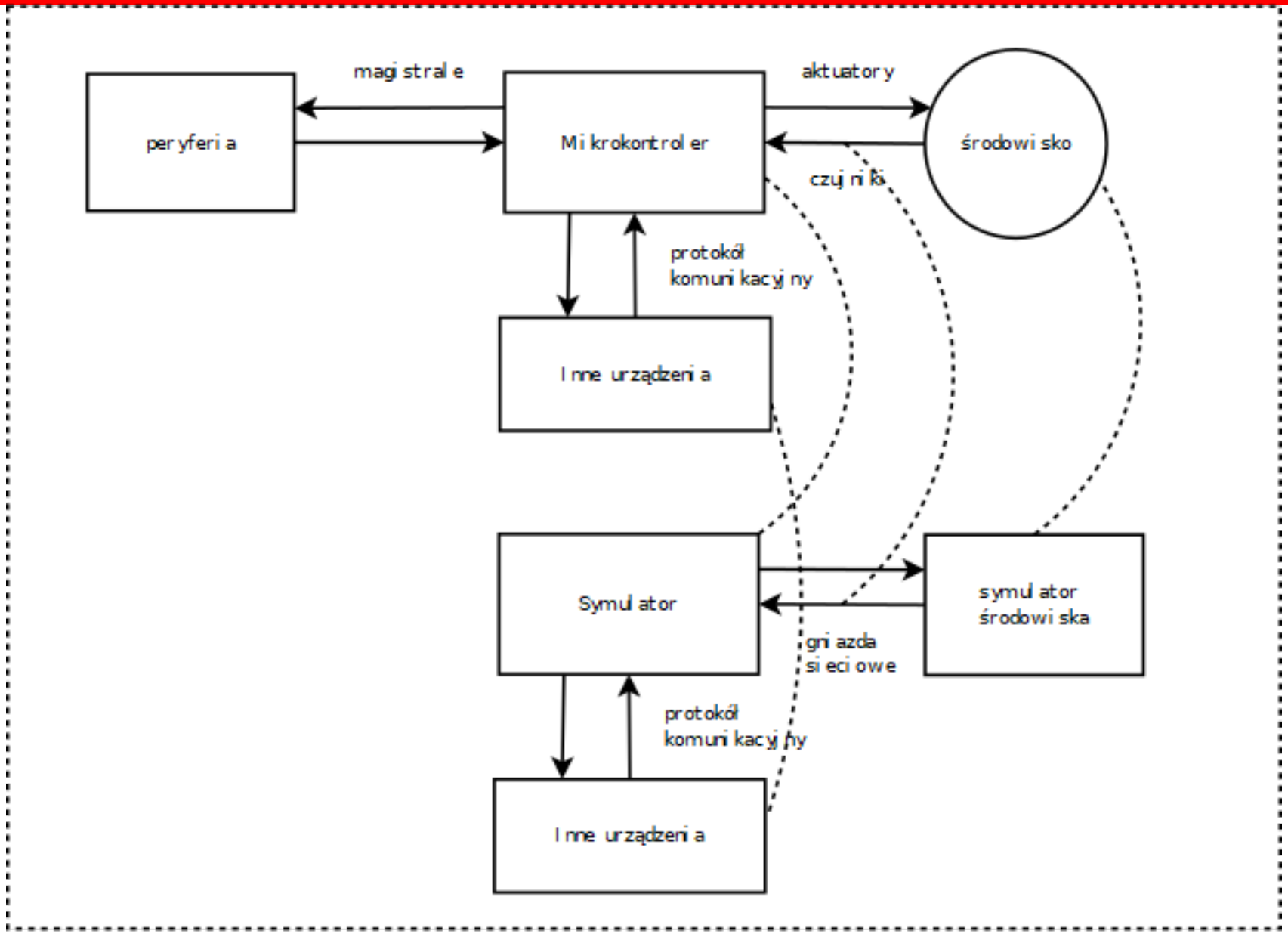
Struktura projektu embedded



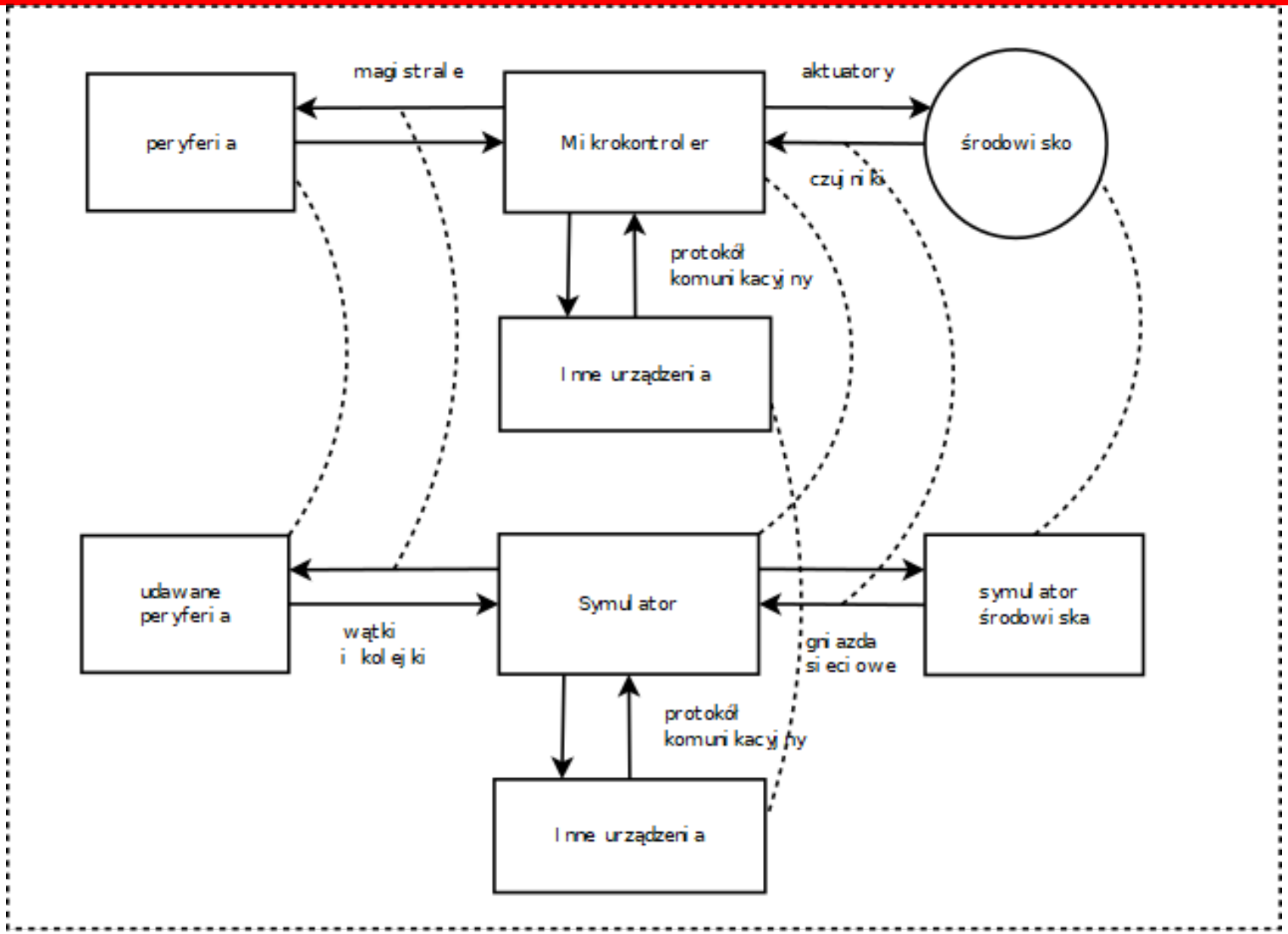
Struktura projektu embedded



Struktura projektu embedded



Struktura projektu embedded



Symulator środowiska

```
#lang racket
;; https://github.com/panicz/praktyki-embedded
```

Symulator środowiska

```
#lang racket
;; https://github.com/panicz/praktyki-embedded

(require racket/gui/base racket/tcp
         "grand-syntax.rkt" "ground-scheme.rkt"
         "tcp-server.rkt")
```

Symulator środowiska

```
#lang racket
;; https://github.com/panicz/praktyki-embedded

(require racket/gui/base racket/tcp
         "grand-syntax.rkt" "ground-scheme.rkt"
         "tcp-server.rkt")

(define simulator-window (new frame%))
```

Symulator środowiska

```
#lang racket
;; https://github.com/panicz/praktyki-embedded

(require racket/gui/base racket/tcp
         "grand-syntax.rkt" "ground-scheme.rkt"
         "tcp-server.rkt")

(define simulator-window (new frame%))

(define temperature
  (new slider% [label "Temperatura"]
               [min-value -273]
               [max-value 1000]
               [init-value 20]
               [style '(vertical vertical-label)]
               [parent simulator-window]))
```

Symulator środowiska

```
#lang racket
;; https://github.com/panicz/praktyki-embedded

(require racket/gui/base racket/tcp
         "grand-syntax.rkt" "ground-scheme.rkt"
         "tcp-server.rkt")

(define simulator-window (new frame%))

(define temperature
  (new slider% [label "Temperatura"]
               [min-value -273]
               [max-value 1000]
               [init-value 20]
               [style '(vertical vertical-label)]
               [parent simulator-window]))

(send simulator-window show #true)
```

Symulator środowiska

```
#lang racket
;; https://github.com/panicz/praktyki-embedded

(require racket/gui/base racket/tcp
         "grand-syntax.rkt" "ground-scheme.rkt"
         "tcp-server.rkt")

(define simulator-window (new frame%))

(define temperature
  (new slider% [label "Temperatura"]
               [min-value -273]
               [max-value 1000]
               [init-value 20]
               [style '(vertical vertical-label)]
               [parent simulator-window]))

(send simulator-window show #true)

(tcp-server 12345
 ("temperature" (send temperature get-value))
 (_ 0))
```

Korzystanie z gniazd sieciowych

```
#ifdef MINGW
#  define WIN32_LEAN_AND_MEAN
#  include <winsock2.h>
#  include <Ws2tcpip.h>
#  include <errno.h>
#else // !MINGW
#  include <sys/types.h>
#  include <sys/socket.h>
#  include <sys/select.h>
#  include <netinet/in.h>
#  include <netinet/tcp.h>
#  include <arpa/inet.h>
#endif // !MINGW
#include <stdio.h>
#include <assert.h>
#include <stdlib.h>
#include <stdint.h>
#include <ctype.h>
#include <string.h>
```

Korzystanie z gniazd sieciowych

```
int main() {
#ifdef MINGW
  WSADATA wsadata;
  if (WSAStartup(MAKEWORD(1,1), &wsadata) == SOCKET_ERROR) {
    puts("Failed to initialize winsock subsystem");
    return -1;
  }
#endif // MINGW
```

Korzystanie z gniazd sieciowych

```
int main() {
#ifdef MINGW
  WSADATA wsadata;
  if (WSAStartup(MAKEWORD(1,1), &wsadata) == SOCKET_ERROR) {
    puts("Failed to initialize winsock subsystem");
    return -1;
  }
#endif // MINGW
  int sock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
  assert(sock >= 0);

  struct sockaddr_in address;
  address.sin_addr.s_addr = inet_addr("127.0.0.1");
  address.sin_family = AF_INET;
  address.sin_port = htons(12345);

  int error = connect(sock, (struct sockaddr *) &address,
                      sizeof(address));
  assert(!error);
```

Korzystanie z gniazd sieciowych

```
  char line[255];
  char response[255];
  while (1) {
    fgets(line, sizeof(line), stdin);
    int sent = send(sock, line, strlen(line), 0);
    int received = recv(sock, response,
                        sizeof(response), 0);
    response[received] = '\0';
    printf("%s", response);
    fflush(stdout);
  }
  return 0;
}
```

Makefile

```
HOST_OS = $(shell uname -o)

ifeq ($(HOST_OS), Msys)
  LIBS = -mwindows -lws2_32 -lmingw32 -DMINGW
endif

all: tcpclient

tcpclient: cl.c
    gcc $(CFLAGS) $< -o $@ $(LIBS)

clean:
    rm tcpclient
```

W docelowym kodzie

```
#include "config.h" // definiuje stale DEVICE i SIMULATOR

int current_temperature_C(void) {
  int temperature_C = 20;
```

W docelowym kodzie

```
#include "config.h" // definiuje stale DEVICE i SIMULATOR

int current_temperature_C(void) {
  int temperature_C = 20;
#if TARGET == SIMULATOR
  int n = send(Global.socket, "temperature",
               sizeof("temperature"), 0);
  assert(n == sizeof("temperature"));
  char buffer[16];
  n = recv(Global.socket, buffer, sizeof(buffer), 0);
  assert(n > 0 && n < 16);
  buffer[n] = '\0';
  sscanf(buffer, "%d", &temperature_C);
```

W docelowym kodzie

```
#include "config.h" // definiuje stale DEVICE i SIMULATOR

int current_temperature_C(void) {
  int temperature_C = 20;
#if TARGET == SIMULATOR
  int n = send(Global.socket, "temperature",
               sizeof("temperature"), 0);
  assert(n == sizeof("temperature"));
  char buffer[16];
  n = recv(Global.socket, buffer, sizeof(buffer), 0);
  assert(n > 0 && n < 16);
  buffer[n] = '\0';
  sscanf(buffer, "%d", &temperature_C);
#elif TARGET == DEVICE
  i2c_read(temperatureSensor, &temperature_C);
```

W docelowym kodzie

```
#include "config.h" // definiuje stale DEVICE i SIMULATOR

int current_temperature_C(void) {
  int temperature_C = 20;
#if TARGET == SIMULATOR
  int n = send(Global.socket, "temperature",
               sizeof("temperature"), 0);
  assert(n == sizeof("temperature"));
  char buffer[16];
  n = recv(Global.socket, buffer, sizeof(buffer), 0);
  assert(n > 0 && n < 16);
  buffer[n] = '\0';
  sscanf(buffer, "%d", &temperature_C);
#elif TARGET == DEVICE
  i2c_read(temperatureSensor, &temperature_C);
#else
#  error "Unknown target"
#endif // TARGET
  return temperature_C;
}
```

Sterownik wyświetlacza - struktura

```
struct {
  // ...
#if TARGET==SIMULATOR
  SDL_Window *window;
  SDL_Surface *surface;
  pthread_t sync_screen;
  uint32_t LTDC_CLUT[256];
  int socket;
#endif // TARGET==SIMULATOR
  // ...
  uint8_t videobuffer[SCREEN_HEIGHT][SCREEN_WIDTH];
  // ...
} Global;
```

Sterownik wyświetlacza - inicjalizacja

```
int main(void) {
// ...
#if TARGET == SIMULATOR
  SDL_Init(SDL_INIT_VIDEO);
  Global.window =
    SDL_CreateWindow("embedded device",
                     SDL_WINDOWPOS_UNDEFINED,
                     SDL_WINDOWPOS_UNDEFINED,
                     SCREEN_WIDTH, SCREEN_HEIGHT,
                     0);
  Global.surface = SDL_GetWindowSurface(Global.window);
  assert(Global.surface);
  pthread_create(&Global.sync_screen, NULL,
                 LTDC_SDL_sync, NULL);
#endif // TARGET == SIMULATOR
  // ...
}
```

Sterownik wyświetlacza - symulacja

```
void *LTDC_SDL_sync(void *unused) {
  while (1) {
    uint32_t *pixels = (uint32_t *) Global.surface->pixels;
    for (int line = 0; line < SCREEN_HEIGHT; ++line) {
      for (int pixel = 0; i < SCREEN_WIDTH; ++pixel) {
            pixels[line*SCREEN_WIDTH + pixel] =
              Global.LTDC_CLUT[
                Global.videobuffer[line*SCREEN_WIDTH + pixel]
              ];
      }
      SDL_UpdateWindowSurface(window);
      if (line == sync_line) {
        LTDC_LineEvent();
      }
    }
  }
  return NULL;
}
```

Przykładowa aplikacja (pętla główna)

```
    while (1) {
      int temperature_C = current_temperature_C();
      uint8_t color;
      if (temperature_C < -16) {
        color = 0;
      }
      else if (temperature_C < 145) {
        color = (uint8_t) (temperature_C + 16);
      }
      else {
        color = 255;
      }
      fill_rect(0, 0, SCREEN_WIDTH, SCREEN_HEIGHT, color);
#if TARGET==SIMULATOR
      SDL_Event event;
      while (SDL_PollEvent(&event)) {
        if (event.type == SDL_QUIT) {
          exit(0);
        }
      }
#endif // TARGET==SIMULATOR
```

Koniec

Pytania?

https://github.com/panicz/praktyki-embedded
https://github.com/panicz/pamphlet
https://www.quora.com/profile/Panicz-Godek
godek.maciek@gmail.com
@PaniczGodek