Vhodno-izhodne naprave (VIN)

Predavanja

11. Načrtovanje in programiranje vgrajenih sistemov – teorija in praksa

Robert Rozman rozman@fri.uni-lj.si

Vsebina

1. Načrtovanje vgrajenih sistemov (HW, SW, …)

2. Programiranje vgrajenih sistemov – primeri:

- Cubesensors, Tevel, D13, Tinia, …

3. Nivoji programiranja

- baremetal (zbirnik,C), HAL knjižnica, ena zanka, končni avtomati, RTOS

4. Podrobnejši primeri programiranja – RTOS

4.1 Splošno o RTOS

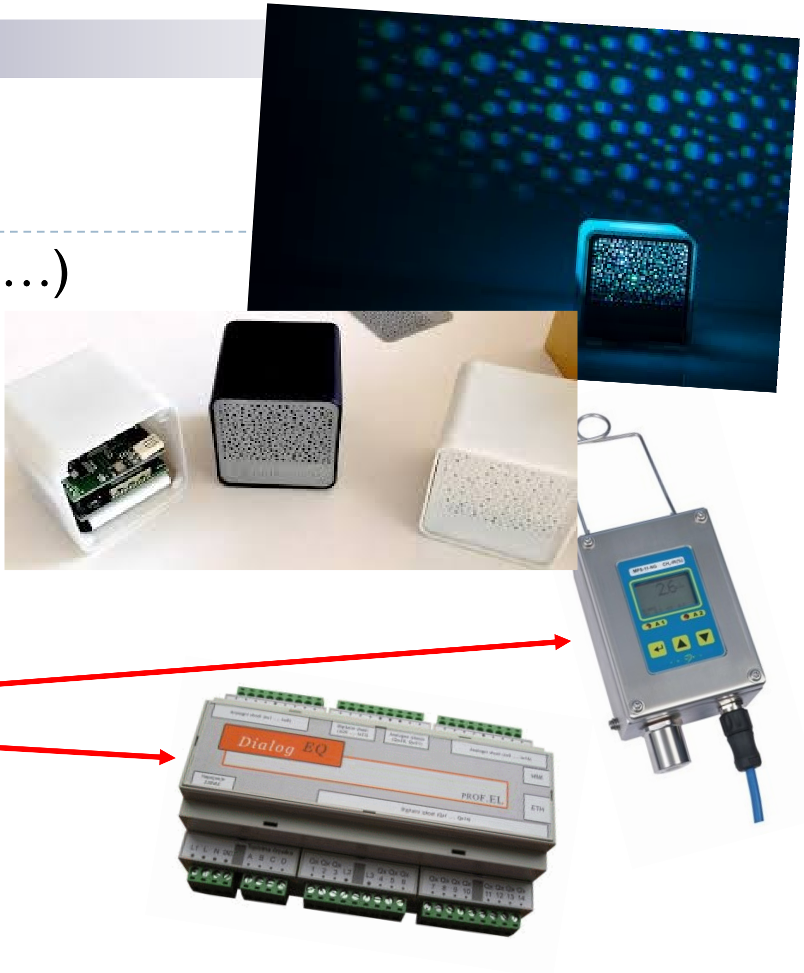
4.2 FreeRTOS

4.3 MQX RTOS

Vsebina I:

1. Načrtovanje vgrajenih sistemov (HW, SW, …)

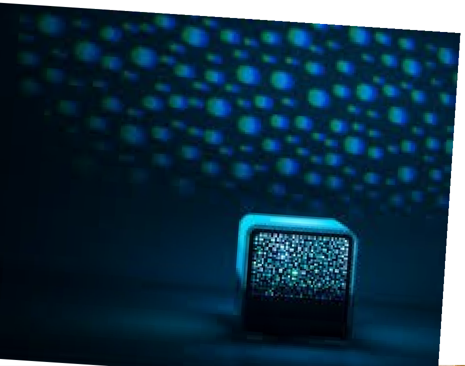
2. Programiranje vgrajenih sistemov :
   - splošen pogled
   - primeri :
     - A. Cubesensors („pametne kocke“)
       - □ realizacija v enotni zanki, končni avtomat
     - B. Tevel – univerzalni merilniki (ekspl. cona)
     - C. D13 („pametni hišni regulator“)
       - □ RTOS (primer MQX)
     - D. Tinia – Prijazen dom
     - E. Pametni zabojnik
     - F. Embedded Linux (UcLinux, Buildroot)
     - G. Simulacije (QEMU)
   - CubeIDE: razvoj in razhroščevanje
   - Kaj po razvoju ? Skrb za stabilnost sistemov v praksi
     - preventiva in kurativa



Vsebina II:

3. Nivoji programiranja

```
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Jan  1 2014, 17:13:19.
Port /dev/ttyUSB1, 10:37:40

Press CTRL-A Z for help on special keys

Here are
```



4. Podrobnejši primeri programiranja – RTOS

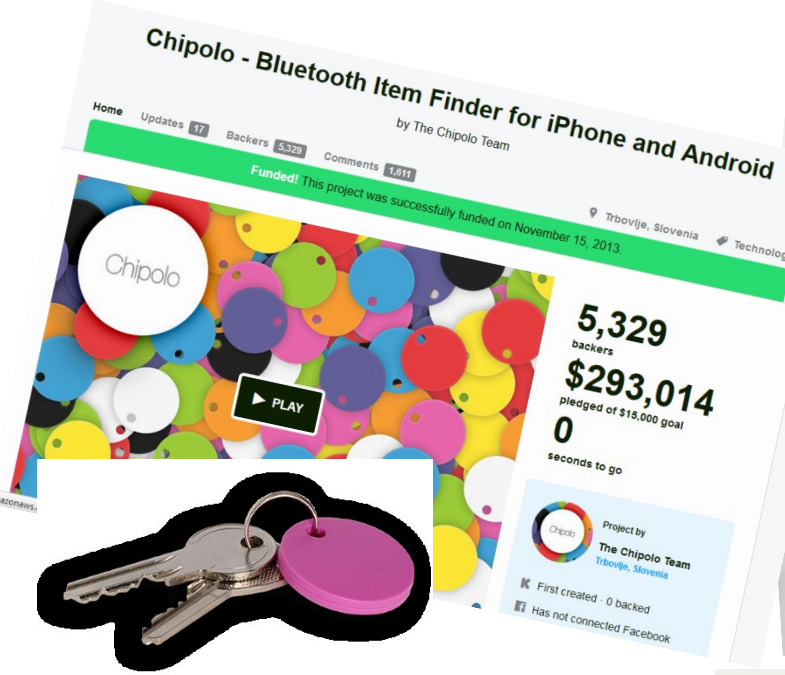
   4.1 Splošno o RTOS

   4.2 FreeRTOS

   4.3 MQX RTOS

```
void mytask(uint_32 startup_parameter)  {
   /* Task initialization code */
   ....
   while (1) {
      /* Task body */
      ....
      ....
   }

}
```

Zakaj HW (in SW) ?



Chipolo - Bluetooth Item Finder for iPhone and Android

by The Chipolo Team

Home Updates 17 Backers 5,329 Comments 1,611

Funded! This project was successfully funded on November 15, 2013.

Trbovlje, Slovenia Technology

5,329 backers

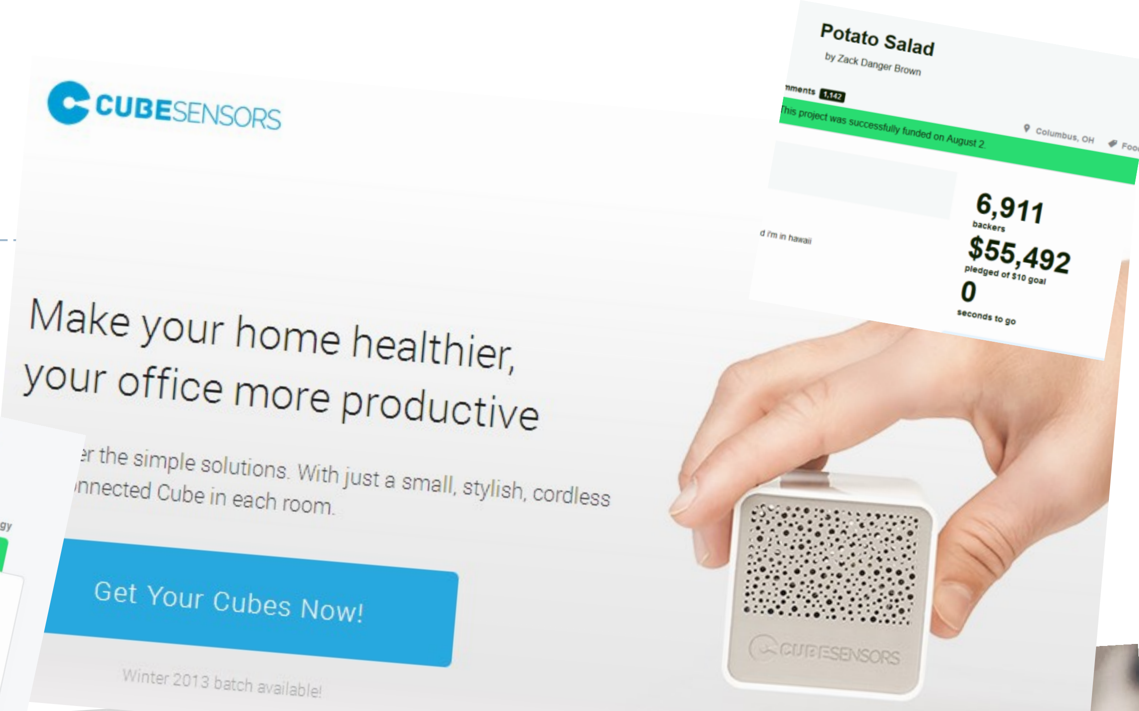
$293,014 pledged of $15,000 goal

0 seconds to go

Project by The Chipolo Team Trbovlje, Slovenia

First created · 0 backed

Has not connected Facebook



CUBESENSORS

Make your home healthier, your office more productive

r the simple solutions. With just a small, stylish, cordless nnected Cube in each room.

Get Your Cubes Now!

Winter 2013 batch available!

Potato Salad

by Zack Danger Brown

mments 1,142

This project was successfully funded on August 2

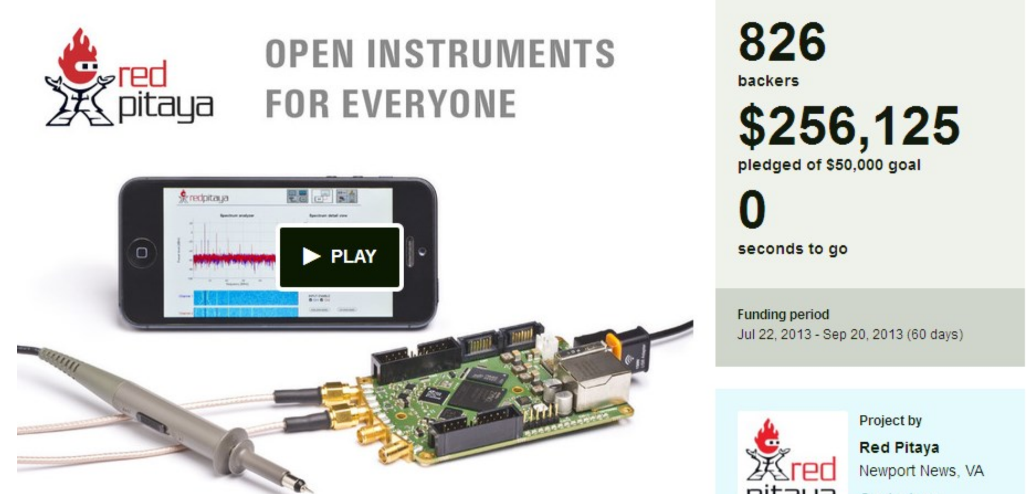
Columbus, OH Foo

d i'm in hawaii

6,911 backers

$55,492 pledged of $10 goal

0 seconds to go



red pitaya OPEN INSTRUMENTS FOR EVERYONE

826 backers

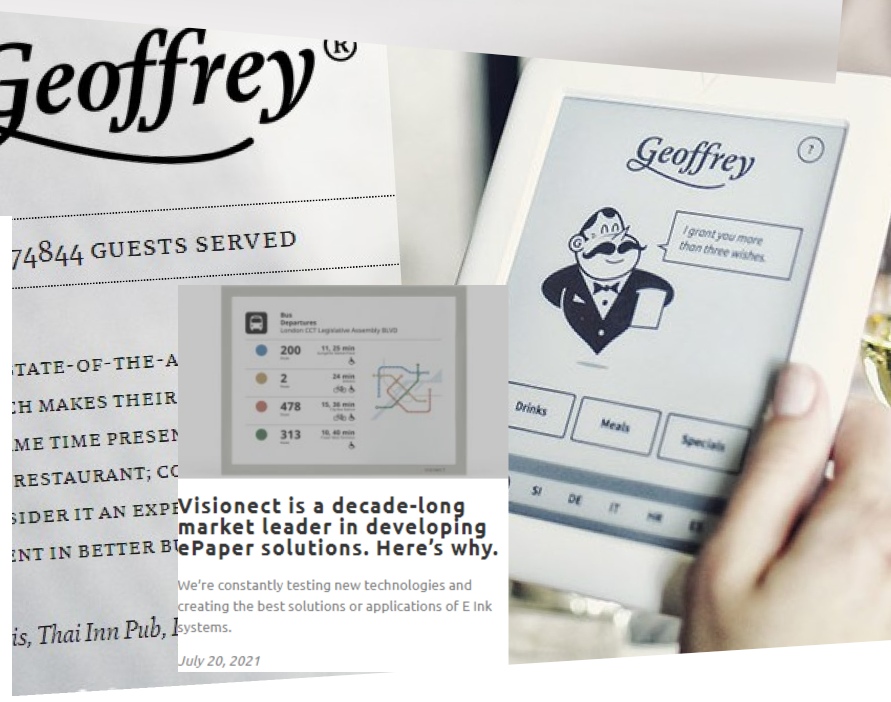
$256,125 pledged of $50,000 goal

0 seconds to go

Funding period

Jul 22, 2013 - Sep 20, 2013 (60 days)

Project by Red Pitaya Newport News, VA



Geoffrey®

74844 GUESTS SERVED

TATE-OF-THE-A
CH MAKES THEIR
ME TIME PRESEN
RESTAURANT; CO
SIDER IT AN EXP
ENT IN BETTER B

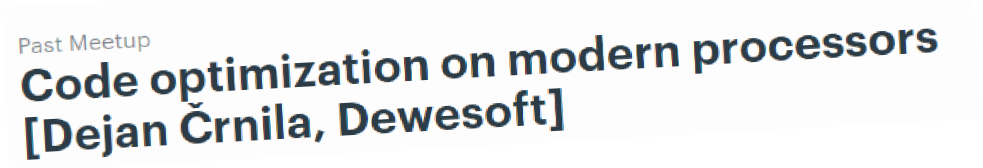
is, Thai Inn Pub, I

Visionect is a decade-long market leader in developing ePaper solutions. Here's why.

We're constantly testing new technologies and creating the best solutions or applications of E Ink systems.

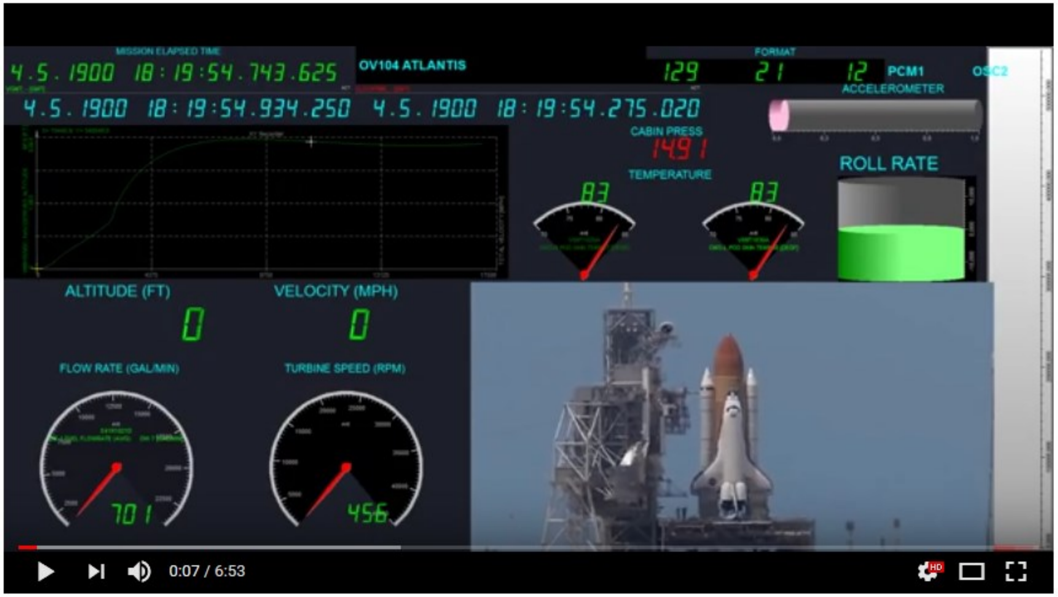
July 20, 2021

Zakaj HW (in SW) ?



Past Meetup

Code optimization on modern processors [Dejan Črnila, Dewesoft]



Space shuttle Atlantis launch monitoring with Dewesoft software

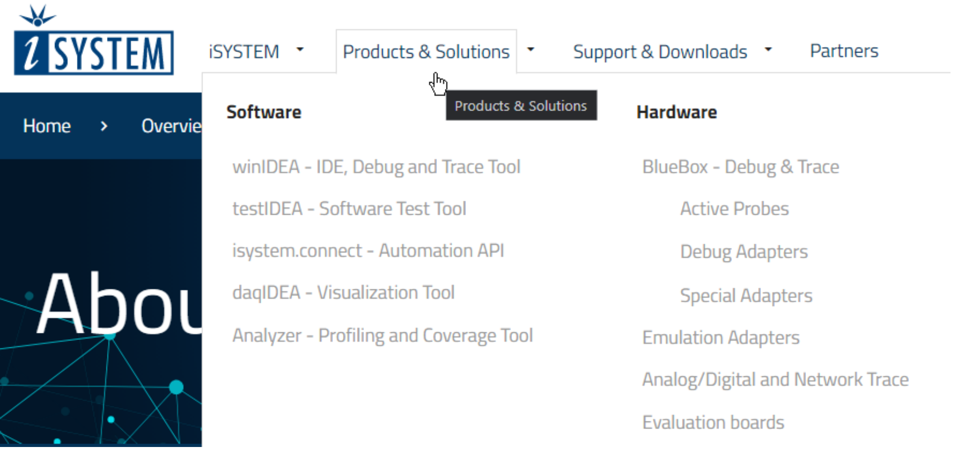
SMARTEH LIVING SYSTEMS

PRODUKTI IN REŠITVE ZA PAMETNO AVTOMATIZACIJO ZGRADB

RAZIŠČI

Majhen Koaksialni Helikopter SCH-2A

Najlažji osebni koaksialni helikopter na svetu...



iSYSTEM

iSYSTEM | Products & Solutions | Support & Downloads | Partners

Home > Overvie

Products & Solutions

Software

- winIDEA - IDE, Debug and Trace Tool
- testIDEA - Software Test Tool
- isystem.connect - Automation API
- daqIDEA - Visualization Tool
- Analyzer - Profiling and Coverage Tool

Hardware

- BlueBox - Debug & Trace
  - Active Probes
  - Debug Adapters
  - Special Adapters
- Emulation Adapters
- Analog/Digital and Network Trace
- Evaluation boards

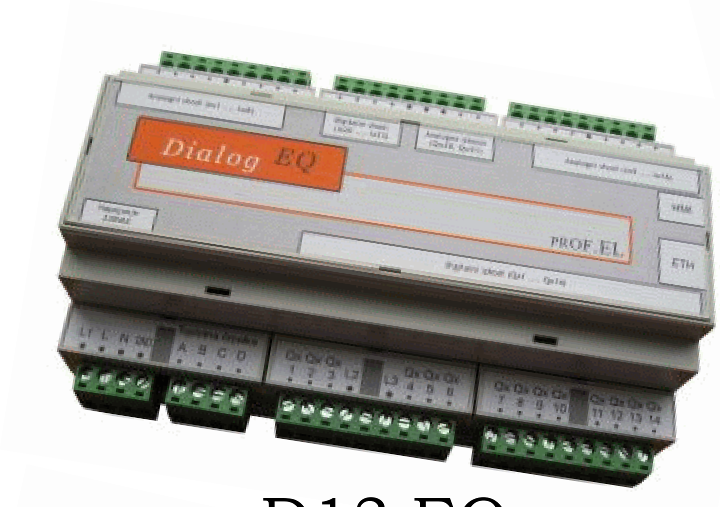
Abou

Trapview STANDARD

Most widely deployed model of the trap has a similar external appearance as the conventional delta traps, therefore its pest catch efficiency is on the same level with conventional traps as well.

Prikaz primerov vgrajenih sistemov

FRI-SMS



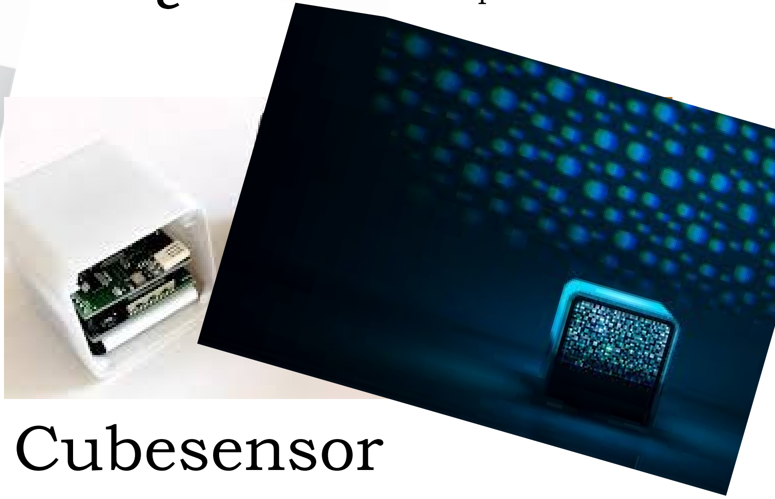
D13 EQ

Tevel

Merilnik konc. plinov

STM Discovery

Tempmate S1



Cubesensor

1. Načrtovanje vgrajenih sistemov

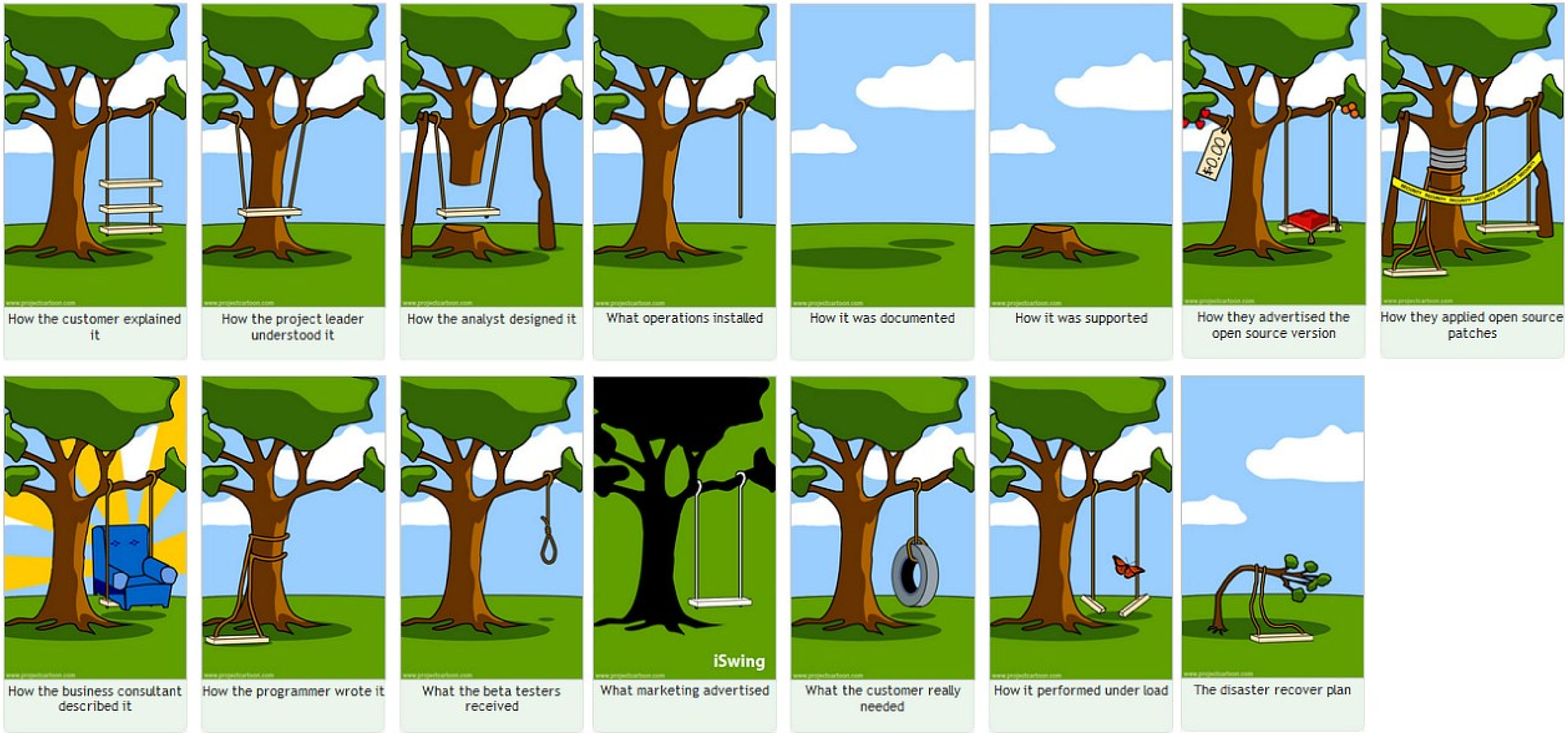
Običajen potek :

- Specifikacija – opredelitev zahtev ->
  - zelo pomembna !
- Izbira el. komponent, čipov, krmilnikov, itd… ->
  - pregled dokumentacije („Errata“, rok dobavljivosti, podpora,…)
- Načrtovanje PCB
- Prvi zagon – oživljanje sistema, razvoj SW
- Spremljanje delovanja

1. Načrtovanje vgrajenih sistemov

- Specifikacija – opredelitev zahtev
  - zelo pomembna !

Product development from an IT failures perspective



How the customer explained it

How the project leader understood it

How the analyst designed it

What operations installed

How it was documented

How it was supported

How they advertised the open source version

How they applied open source patches

How the business consultant described it

How the programmer wrote it

What the beta testers received

What marketing advertised

What the customer really needed

How it performed under load

The disaster recover plan

1. Načrtovanje vgrajenih sistemov

Izbira el. komponent, čipov, krmilnikov, itd…

- Datasheet (DS):
  - „kako bi naj delovalo…“
- Errata:
  - „kaj vse ne deluje tako kot v DS…“
- Rok dobavljivosti
- Podpora ->

Prazna ?

Pentium FDIV bug:

The **Pentium FDIV bug** is a bug in the Intel P5 Pentium floating point unit (FPU). Because of the bug, the processor can return incorrect decimal results, an issue troublesome for the precise calculations needed in fields like math and science.

Errata : + dobro, vsaj znan problem

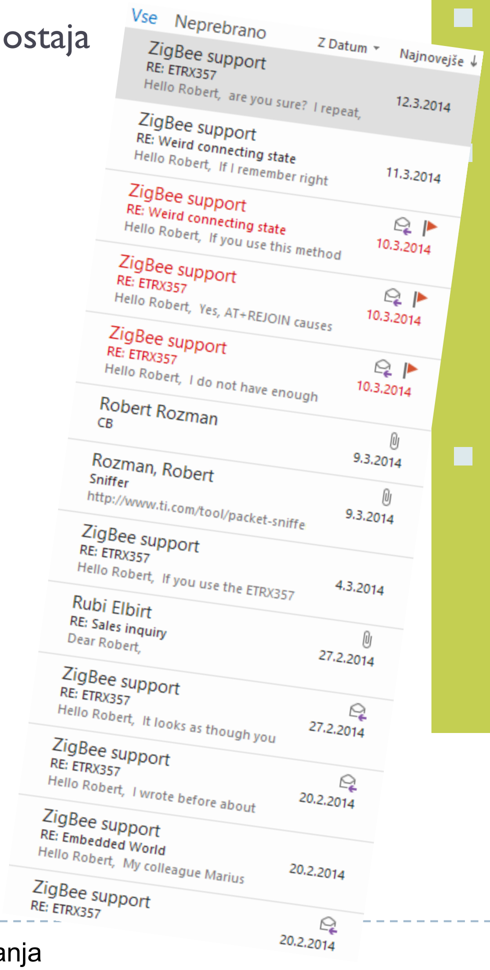
Kaj če naletimo na neznan problem?

- ***upamo na reprodukcijo in podporo***

Izbira el. komponent, čipov, krmilnikov, itd… - 2 zgodbi

Podpora - 1. zgodba :

- težave z brezžičnim modulom
  - cca. 70 emailov,
  - problem ostaja



Vse Neprebrano Z Datum ▾ Najnovejše ↓

ZigBee support
RE: ETRX357
Hello Robert, are you sure? I repeat,
12.3.2014

ZigBee support
RE: Weird connecting state
Hello Robert, If I remember right
11.3.2014

ZigBee support
RE: Weird connecting state
Hello Robert, If you use this method
10.3.2014

ZigBee support
RE: ETRX357
Hello Robert, Yes, AT+REJOIN causes
10.3.2014

ZigBee support
RE: ETRX357
Hello Robert, I do not have enough
10.3.2014

Robert Rozman
CB
9.3.2014

Rozman, Robert
Sniffer
http://www.ti.com/tool/packet-sniffe
9.3.2014

ZigBee support
RE: ETRX357
Hello Robert, If you use the ETRX357
4.3.2014

Rubi Elbirt
RE: Sales inquiry
Dear Robert,
27.2.2014

ZigBee support
RE: ETRX357
Hello Robert, It looks as though you
27.2.2014

ZigBee support
RE: ETRX357
Hello Robert, I wrote before about
20.2.2014

ZigBee support
RE: Embedded World
Hello Robert, My colleague Marius
20.2.2014

ZigBee support
RE: ETRX357
20.2.2014

Zaznani hrošči:

- **»stuck in error94«**

  Main.c : / **hack when zigbee module gets stuck** in error 94. nothing except AT&F resolves the issue, no ATZ, reset pin, power down */*

- **»no SEQ prompt«**

  / Bits are set to 1, when message is in air (SEQ, but not yet ACK or NACK). */*
  / **In a perfect world this would not be needed**, but it seems like module*
  ** sometimes (RARELY) does not send SEQ: after AT+UCAST*, but it does send ACK:*
  ** afterwards. If that happens, pending_messages buffer can become -1 long,*
  ** and old (or not yet used, invalid) messages are sent. This is a suspect*
  ** for ticket:185 */*

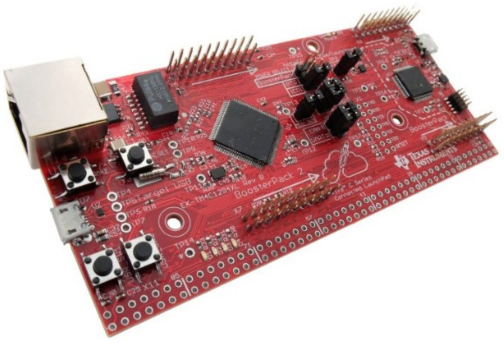
- **»not be able to reconnect »**

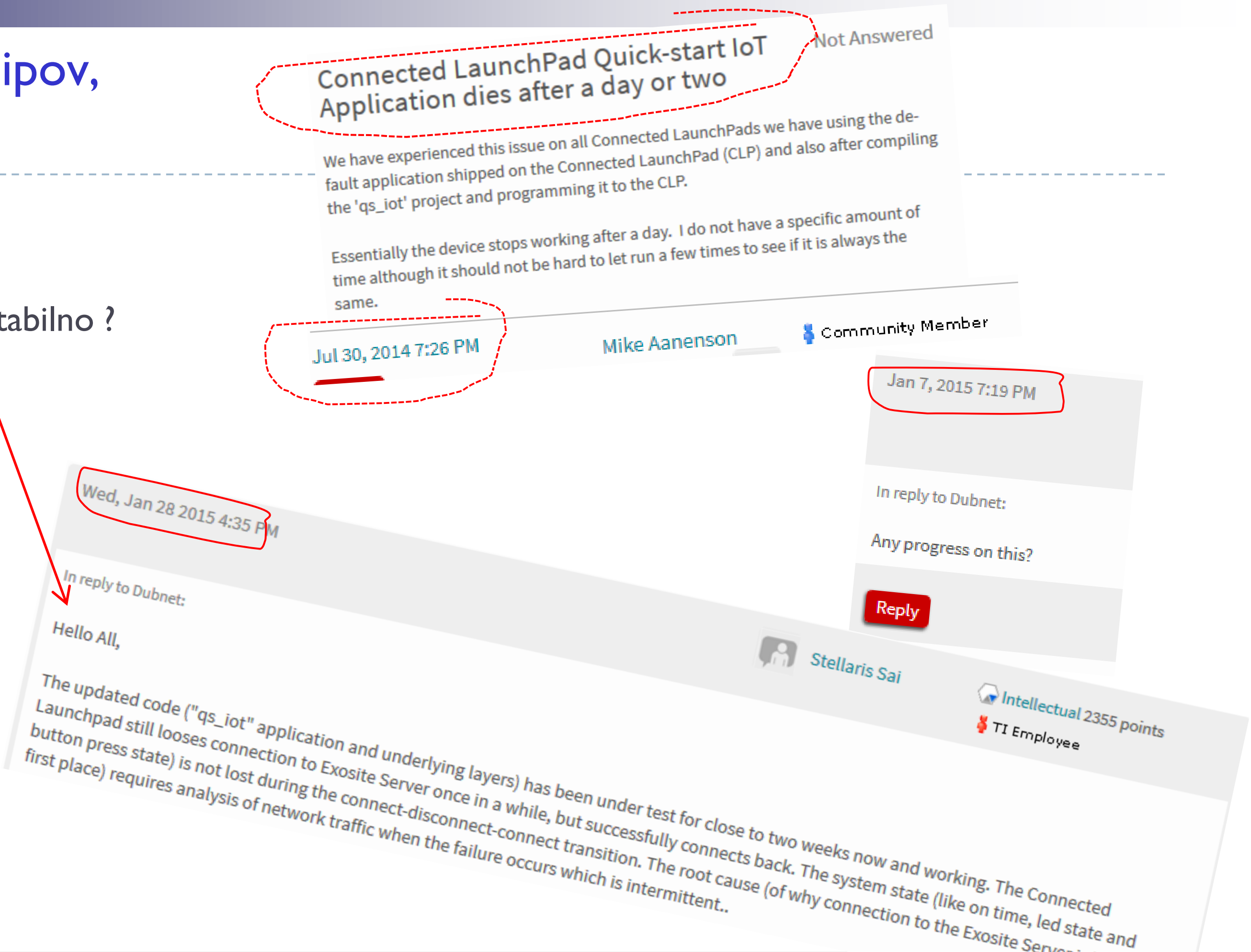
  static void check_initialized(void) {
  int r;
  / zigbee module bug workaround.*
  ** **Reset the module, or it will not be able to reconnect** after seeing*
  ** a bunch of modules failing to connect (NEWNODE w/o FFD).*
  **/*

Izbira el. komponent, čipov, krmilnikov, itd…

Podpora - 2. zgodba:

- Connected launchpad
  - nov izdelek – ne deluje stabilno ?
  - v enem letu ni rešitve





Connected LaunchPad Quick-start IoT Application dies after a day or two

Not Answered

We have experienced this issue on all Connected LaunchPads we have using the default application shipped on the Connected LaunchPad (CLP) and also after compiling the 'qs_iot' project and programming it to the CLP.

Essentially the device stops working after a day. I do not have a specific amount of time although it should not be hard to let run a few times to see if it is always the same.

Jul 30, 2014 7:26 PM | Mike Aanenson | Community Member

Jan 7, 2015 7:19 PM

In reply to Dubnet:

Any progress on this?

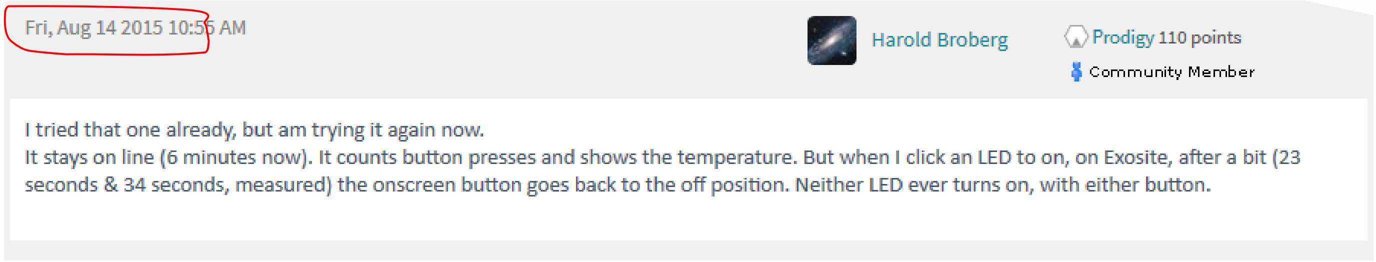
Reply

Wed, Jan 28 2015 4:35 PM | Stellaris Sai | Intellectual 2355 points | TI Employee

In reply to Dubnet:

Hello All,

The updated code ("qs_iot" application and underlying layers) has been under test for close to two weeks now and working. The Connected Launchpad still looses connection to Exosite Server once in a while, but successfully connects back. The system state (like on time, led state and button press state) is not lost during the connect-disconnect-connect transition. The root cause (of why connection to the Exosite Server is lost in the first place) requires analysis of network traffic when the failure occurs which is intermittent..



Fri, Aug 14 2015 10:55 AM | Harold Broberg | Prodigy 110 points | Community Member

I tried that one already, but am trying it again now.
It stays on line (6 minutes now). It counts button presses and shows the temperature. But when I click an LED to on, on Exosite, after a bit (23 seconds & 34 seconds, measured) the onscreen button goes back to the off position. Neither LED ever turns on, with either button.

2. Programiranje vgrajenih sistemov

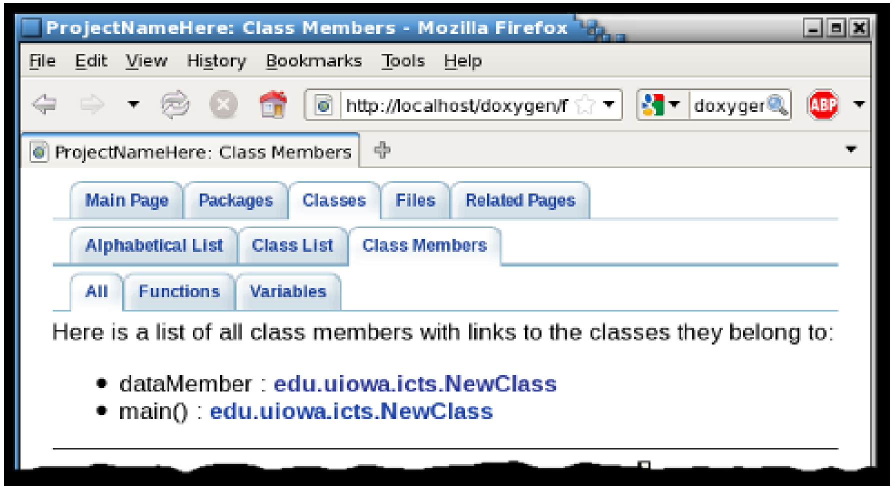
```
pm: Default power policy is performance mode
Initializing Zero device
Initializing Realtime Clock
Initializing Serial Console
IRQ1 attached priority=1
Initializing RAM disk
RAM disk at 0x8002cbac (79K bytes)
Initializing TTY device
Driver initialized
Loading task: boot
Loading task: proc
Loading task: fs
Loading task: exec
Starting Bootstrap Server
Starting Process Server
Starting File System Server
Starting Exec Server
boot: Mounting file systems
VFS: Mounting ramfs dev= dir=/
VFS: Mounting devfs dev= dir=/dev
VFS: Mounting arfs dev=/dev/ram0 dir=/boot
VFS: Mounting fifofs dev= dir=/fifo
boot: Run init process
Prex version 0.8.1 (arm-integrator)
[prex:/]#
```

Splošno :

- Orodja :
  - IDE: CubeIDE, IAR, Keil, Eclipse
    - Pomembne funkcionalnosti : Debug, Profile, …
  - Doxygen.org :

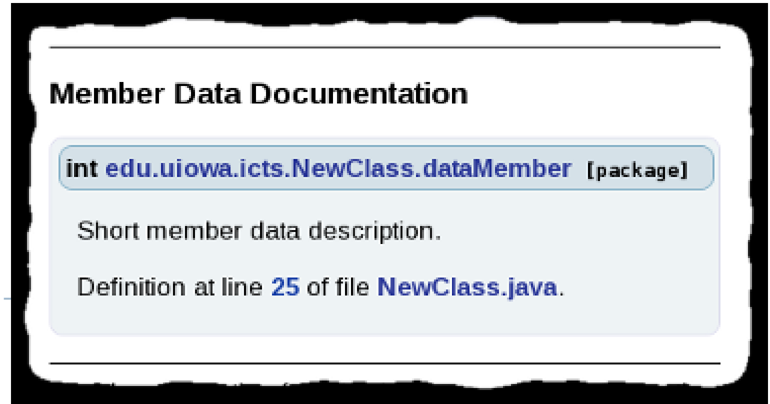
```
/**
 * @brief   Short member data description.
 */
int dataMember;
```

Primer: D13.chm



Here is a list of all class members with links to the classes they belong to:

- dataMember : edu.uiowa.icts.NewClass
- main() : edu.uiowa.icts.NewClass



Member Data Documentation

int edu.uiowa.icts.NewClass.dataMember [package]

Short member data description.

Definition at line 25 of file NewClass.java.

2. Programiranje vgrajenih sistemov

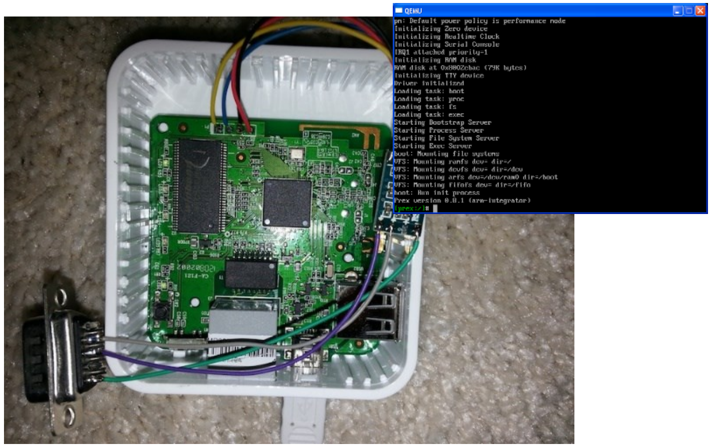
Splošne metode :

razvoj

- Pravila robustnega programiranja
  - MISRA C ->
- Sledenje delovanja programa
  - Debugger

spremljanje

  - Serijska konzola
  - Log datoteka (lokalno)
  - Oddaljen nadzor/logiranje (splet)



```
pm: Default power policy is performance mode
Initializing Zero device
Initializing Realtime Clock
Initializing Serial Console
IRQ1 attached priority=1
Initializing RAM disk
RAM disk at 0x8802cbac (79K bytes)
Initializing TTY device
Driver initialized
Loading task: boot
Loading task: proc
Loading task: fs
Loading task: exec
Starting Bootstrap Server
Starting Process Server
Starting File System Server
Starting Exec Server
boot: Mounting file systems
VFS: Mounting ramfs dev= dir=/
VFS: Mounting devfs dev= dir=/dev
VFS: Mounting arfs dev=/dev/ram0 dir=/boot
VFS: Mounting fifofs dev= dir=/fifo
boot: Run init process
Prex version 0.8.1 (arm-integrator)
[prex:/]#
```

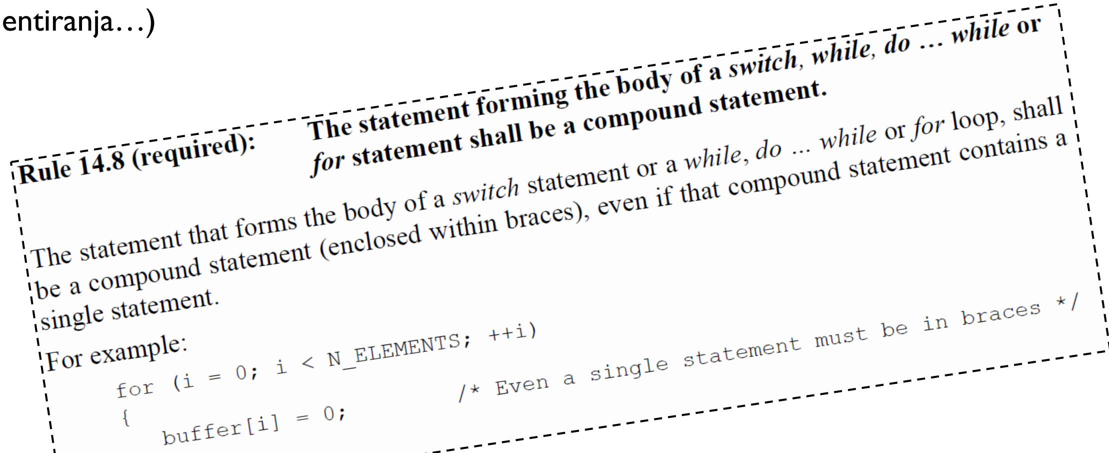
```
2015-01-11 05:45:37 CRIT   232       0 FP       WDT has expired
2015-01-11 05:46:21 CRIT   232       0 MNG      WDT has expired
```

```
2015-01-09 15:00:02 INFO    60       0 CMDEXECUTE  CMD:Execute Cmd[72]
2015-01-09 15:00:02 INFO    60       0 CMDEXECUTE CMD:SendSett
2015-01-09 15:04:02 CRIT   232       0 CMDEXECUTE  WDT has expired
```

2. Programiranje vgrajenih sistemov

Pravila robustnega programiranja (preventiva)

- MISRA C (1998, 2004, 2012):
  - MISRA = **M**otor **I**ndustry **S**oftware **R**eliability **A**ssociation
  - 143 pravil (preverljivih z analizo) in 16 smernic
  - skupine pravil:
    - razlike med prevajalniki (npr. velikost tipa Integer)
    - brez funkcij s pogostejšimi napakami (npr. malloc)
    - obvladljiva koda (pravila imenovanja, komentiranja…)
    - primeri dobre prakse
    - omejitve kompleksnosti
  - že integrirano v nekatera IDE orodja:
    - IAR, Green Hills, …

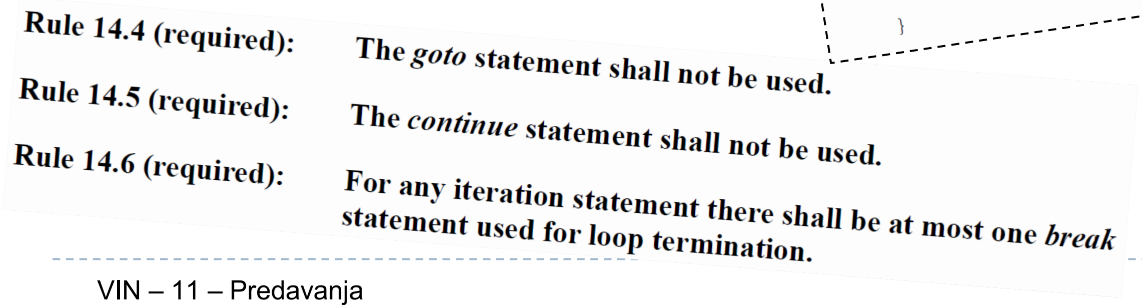


Rule 14.8 (required): **The statement forming the body of a *switch*, *while*, *do … while* or *for* statement shall be a compound statement.**

The statement that forms the body of a *switch* statement or a *while*, *do … while* or *for* loop, shall be a compound statement (enclosed within braces), even if that compound statement contains a single statement.

For example:

```
for (i = 0; i < N_ELEMENTS; ++i)
{                        /* Even a single statement must be in braces */
   buffer[i] = 0;
}
```



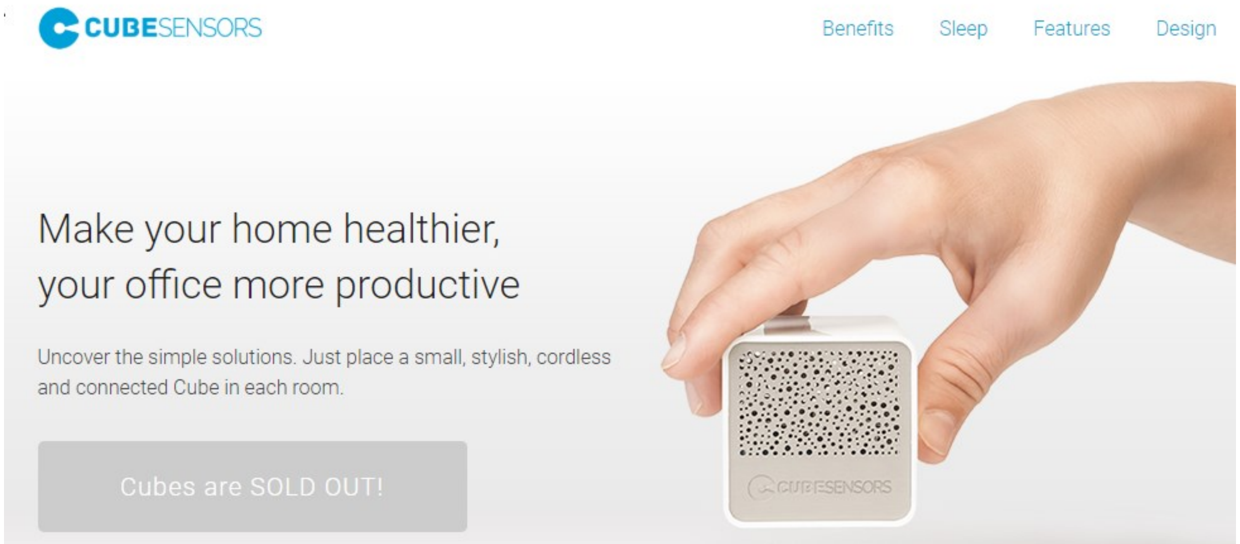
Rule 14.4 (required): **The *goto* statement shall not be used.**

Rule 14.5 (required): **The *continue* statement shall not be used.**

Rule 14.6 (required): **For any iteration statement there shall be at most one *break* statement used for loop termination.**

2. Programiranje vgrajenih sistemov - Primeri

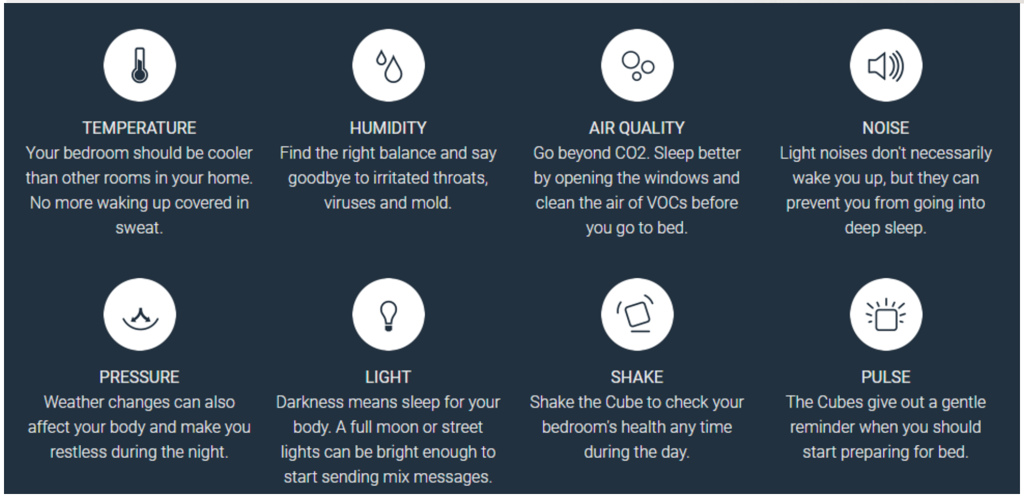
A. Cubesensors („pametne kocke“):



Make your home healthier, your office more productive

Uncover the simple solutions. Just place a small, stylish, cordless and connected Cube in each room.

Cubes are SOLD OUT!



TEMPERATURE
Your bedroom should be cooler than other rooms in your home. No more waking up covered in sweat.

HUMIDITY
Find the right balance and say goodbye to irritated throats, viruses and mold.

AIR QUALITY
Go beyond CO2. Sleep better by opening the windows and clean the air of VOCs before you go to bed.

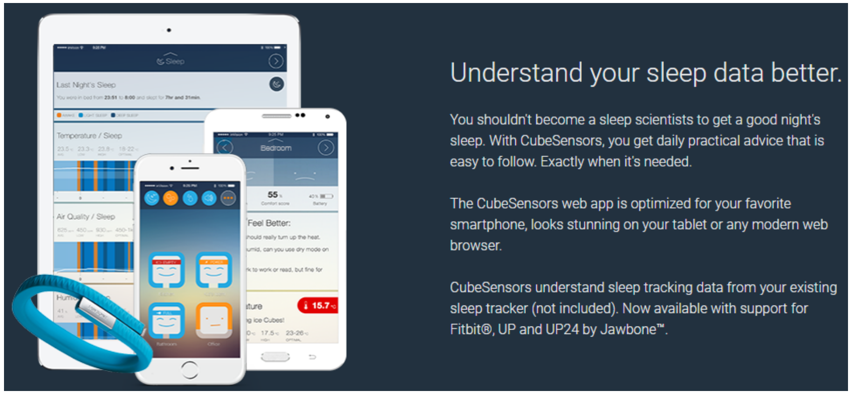
NOISE
Light noises don't necessarily wake you up, but they can prevent you from going into deep sleep.

PRESSURE
Weather changes can also affect your body and make you restless during the night.

LIGHT
Darkness means sleep for your body. A full moon or street lights can be bright enough to start sending mix messages.

SHAKE
Shake the Cube to check your bedroom's health any time during the day.

PULSE
The Cubes give out a gentle reminder when you should start preparing for bed.



Understand your sleep data better.

You shouldn't become a sleep scientists to get a good night's sleep. With CubeSensors, you get daily practical advice that is easy to follow. Exactly when it's needed.

The CubeSensors web app is optimized for your favorite smartphone, looks stunning on your tablet or any modern web browser.

CubeSensors understand sleep tracking data from your existing sleep tracker (not included). Now available with support for Fitbit®, UP and UP24 by Jawbone™.

2. Programiranje vgrajenih sistemov - Prin

<u>A. Cubesensors („pametne kocke“):</u>

- Osnovni model CPU – ARM Cortex M0
- Brezžična komunikacija („Zigbee“)

- Zahteve :
  - nizka poraba, cena
- Odločitev:
  - prehod iz M3 prototipa na M0
- Posledice:
  - omejeno debugiranje
  - ni serijske konzole
  - zelo omejeni viri

2. Programiranje vgrajenih sistemov

A. Cubesensors („pametne kocke“):

Enotna glavna zanka (kompleksnejša izvedba) = **končni avtomat**

- brezžična komunikacija + branje senzorjev + spanje
- boljša organizacija kode, lažje vzdrževanje

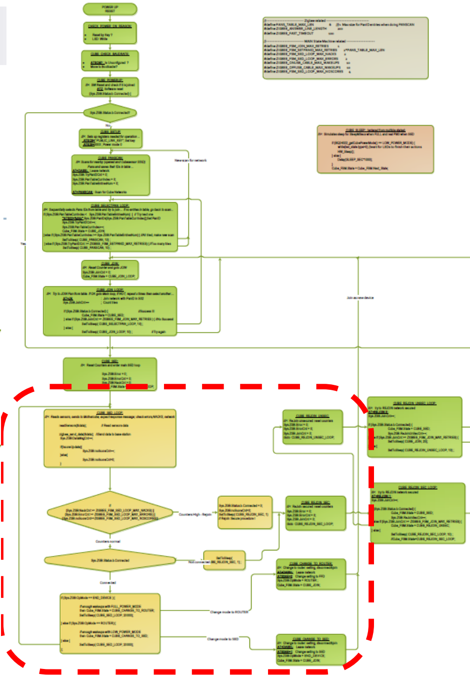
```
switch (FSM.State) {

    case CHECK_POWER_ON_REASON:
            ///<  FSM.State: after reset or power up. SW Reset and check if it can join right away...

          if  VSE_OK then FSM.State  = CHECK_BAUDRATE …

        break;

    case CHECK_BAUDRATE:
            ///<  FSM.State: after reset or power up. SW Reset and check if it can join right away...
          …

        break;
```

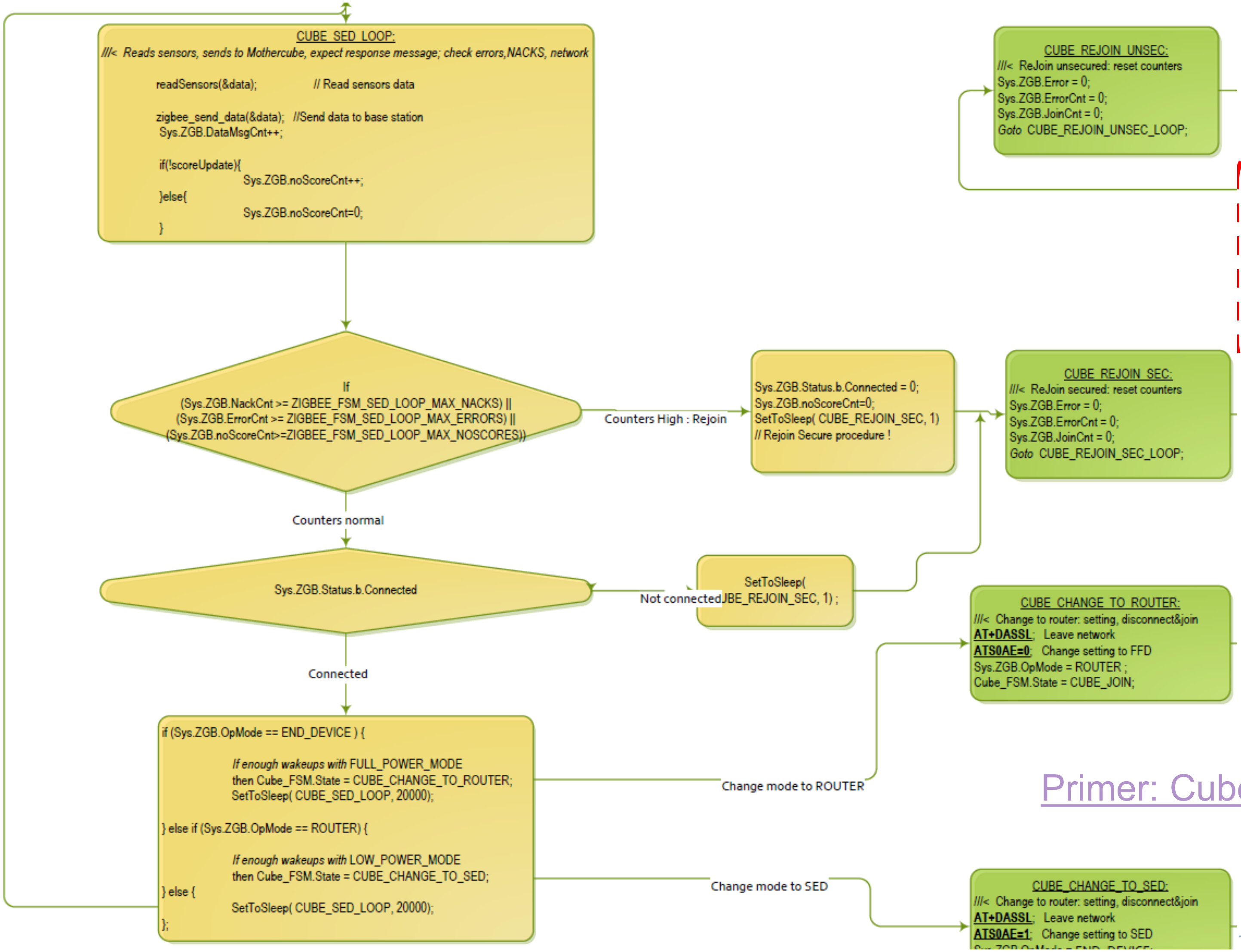
2. Programiranje vgrajenih sistemov

Cubesensors („pametne kocke“):

Diagram poteka (končni avtomat):

- Primer glavnega stanja





CUBE_SED_LOOP:
///< Reads sensors, sends to Mothercube, expect response message; check errors,NACKS, network

```
readSensors(&data);          // Read sensors data

zigbee_send_data(&data);  //Send data to base station
 Sys.ZGB.DataMsgCnt++;

 if(!scoreUpdate){
                    Sys.ZGB.noScoreCnt++;
 }else{
                    Sys.ZGB.noScoreCnt=0;
 }
```

If
(Sys.ZGB.NackCnt >= ZIGBEE_FSM_SED_LOOP_MAX_NACKS) ||
(Sys.ZGB.ErrorCnt >= ZIGBEE_FSM_SED_LOOP_MAX_ERRORS) ||
(Sys.ZGB.noScoreCnt>=ZIGBEE_FSM_SED_LOOP_MAX_NOSCORES))

Counters High : Rejoin

```
Sys.ZGB.Status.b.Connected = 0;
Sys.ZGB.noScoreCnt=0;
SetToSleep( CUBE_REJOIN_SEC, 1)
// Rejoin Secure procedure !
```

CUBE_REJOIN_UNSEC:
///< ReJoin unsecured: reset counters

```
Sys.ZGB.Error = 0;
Sys.ZGB.ErrorCnt = 0;
Sys.ZGB.JoinCnt = 0;
Goto CUBE_REJOIN_UNSEC_LOOP;
```

CUBE_REJOIN_SEC:
///< ReJoin secured: reset counters

```
Sys.ZGB.Error = 0;
Sys.ZGB.ErrorCnt = 0;
Sys.ZGB.JoinCnt = 0;
Goto CUBE_REJOIN_SEC_LOOP;
```

Counters normal

Sys.ZGB.Status.b.Connected

Not connected

SetToSleep(CUBE_REJOIN_SEC, 1) ;

Connected

```
if (Sys.ZGB.OpMode == END_DEVICE ) {

        If enough wakeups with FULL_POWER_MODE
        then Cube_FSM.State = CUBE_CHANGE_TO_ROUTER;
        SetToSleep( CUBE_SED_LOOP, 20000);

} else if (Sys.ZGB.OpMode == ROUTER) {

        If enough wakeups with LOW_POWER_MODE
        then Cube_FSM.State = CUBE_CHANGE_TO_SED;
} else {

        SetToSleep( CUBE_SED_LOOP, 20000);

};
```

Change mode to ROUTER

CUBE_CHANGE_TO_ROUTER:
///< Change to router: setting, disconnect&join

AT+DASSL: Leave network
ATSOAE=0: Change setting to FFD
Sys.ZGB.OpMode = ROUTER ;
Cube_FSM.State = CUBE_JOIN;

Change mode to SED

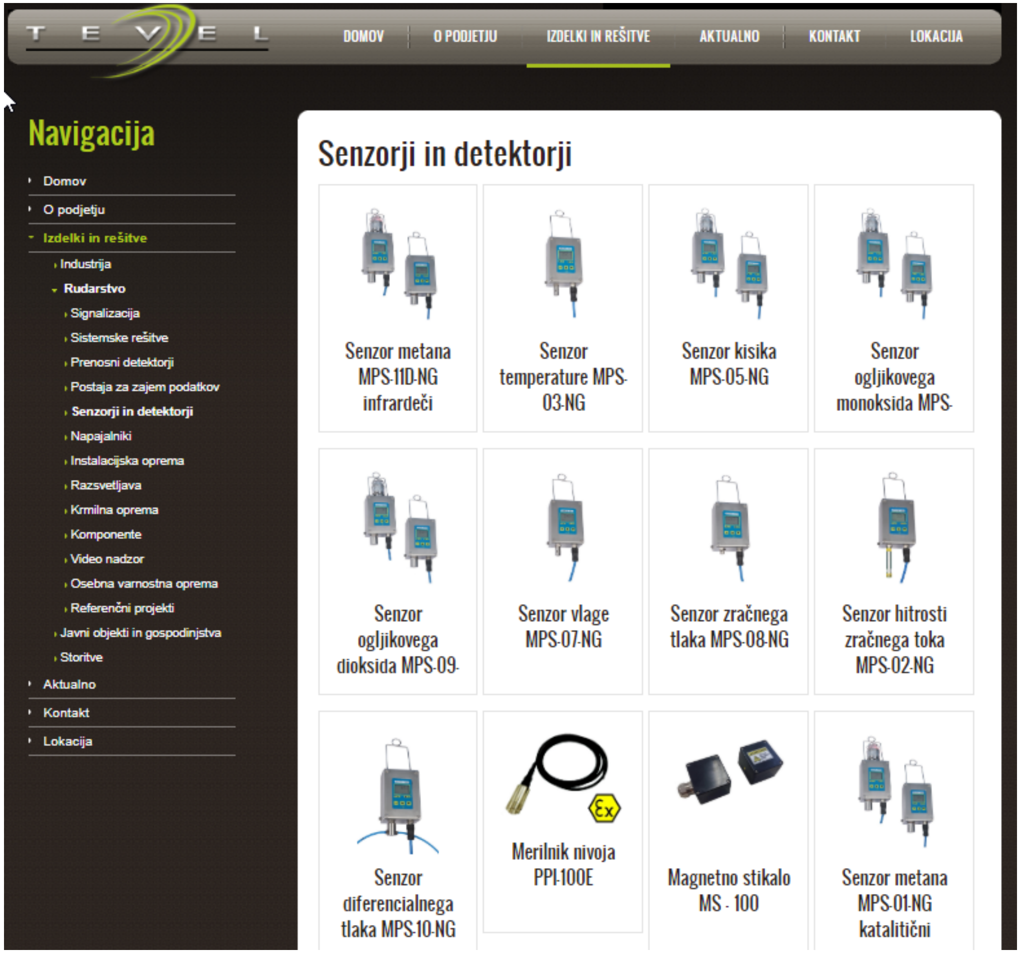
CUBE_CHANGE_TO_SED:
///< Change to router: setting, disconnect&join

AT+DASSL: Leave network
ATSOAE=1: Change setting to SED
Sys.ZGB.OpMode = END_DEVICE;

Primer: Cube_State_Chart_v2.pdf

2. Programiranje vgrajenih sistemov

B. Tevel Pametni merilniki (rudarstvo)



T E V E L

DOMOV | O PODJETJU | IZDELKI IN REŠITVE | AKTUALNO | KONTAKT | LOKACIJA

Navigacija

- Domov
- O podjetju
- Izdelki in rešitve
  - Industrija
  - Rudarstvo
    - Signalizacija
    - Sistemske rešitve
    - Prenosni detektorji
    - Postaja za zajem podatkov
    - Senzorji in detektorji
    - Napajalniki
    - Instalacijska oprema
    - Razsvetljava
    - Krmilna oprema
    - Komponente
    - Video nadzor
    - Osebna varnostna oprema
    - Referenčni projekti
  - Javni objekti in gospodinjstva
  - Storitve
- Aktualno
- Kontakt
- Lokacija

Senzorji in detektorji

Senzor metana MPS-11D-NG infrardeči

Senzor temperature MPS-03-NG

Senzor kisika MPS-05-NG

Senzor ogljikovega monoksida MPS-

Senzor ogljikovega dioksida MPS-09-

Senzor vlage MPS-07-NG

Senzor zračnega tlaka MPS-08-NG

Senzor hitrosti zračnega toka MPS-02-NG

Senzor diferencialnega tlaka MPS-10-NG

Merilnik nivoja PPI-100E

Magnetno stikalo MS - 100

Senzor metana MPS-01-NG katalitični

Tevel v Kazahstanu cilja na rudarsko panogo

(video, foto) Zasavci osvajajo azijske in balkanske rudnike

Gospodarstvo Prva stran

Tevel na Kosovu

10. 10. 2018

Gospodarstvo

Tevel načrtuje nakup nemškega Wölkeja

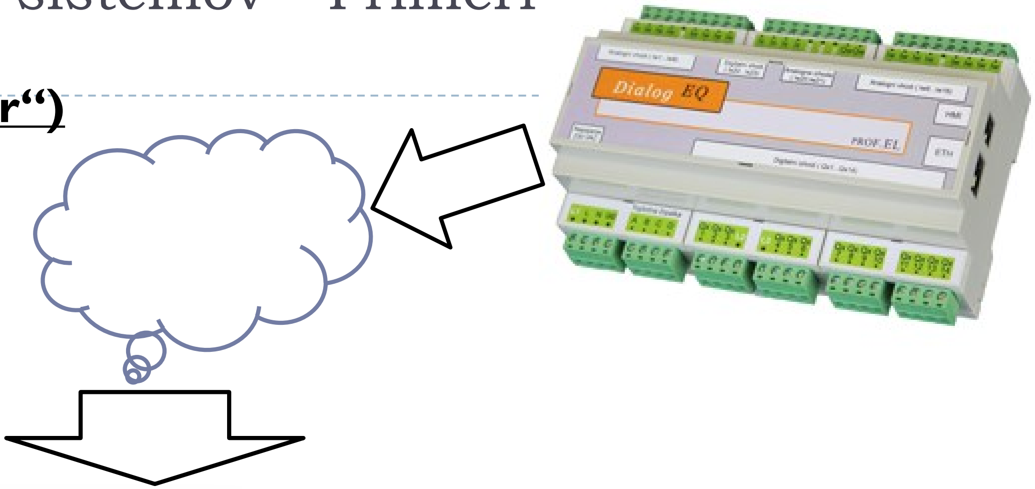
2. Programiranje vgrajenih sistemov - Primeri

B. Tevel Pametni merilniki (rudarstvo):

Enotna glavna zanka – enostavnejša izvedba

```
{ …



        if (Timer_1sec)   {
                readSensors(&data);  // Read sensors
                send_data(&data);      //  Send data to gateway
                 Timer_1sec = 0;
        }

        if (Timer_50msec)   {
                readKeys(&keys);        // Read user keys
                readInputs(&inputs);   // Read digital inputs
                Timer_50msec = 0;
        }

}
```

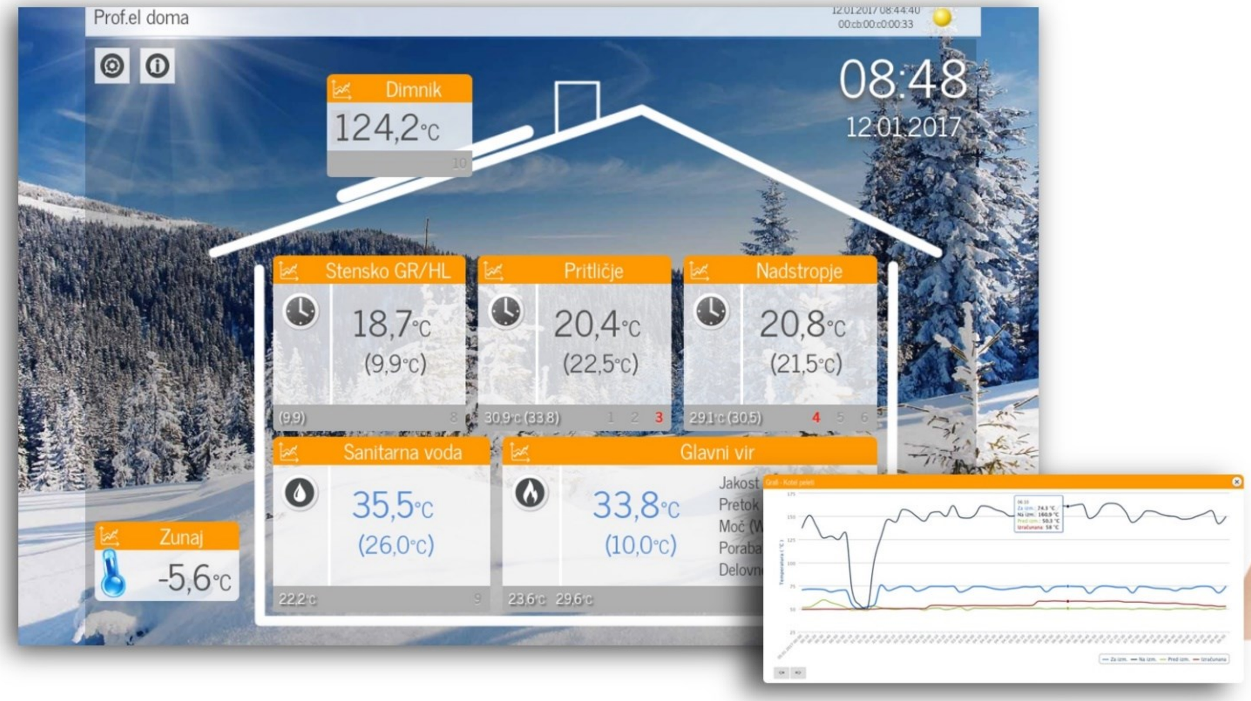
2. Programiranje vgrajenih sistemov - Primeri



C. DIALOG EQ („pametni regulator“)

RTOS – ločeni procesi
(REG,TCP,WEB,MODBUS,CANBUS)

– zahtevnejša izvedba



Prof.el doma

12.01.2017 08:44:40
00:cb:00:c0:00:33

08:48
12.01.2017

Dimnik
124,2°C
10

Stensko GR/HL
18,7°C
(9,9°C)
(9,9) 8

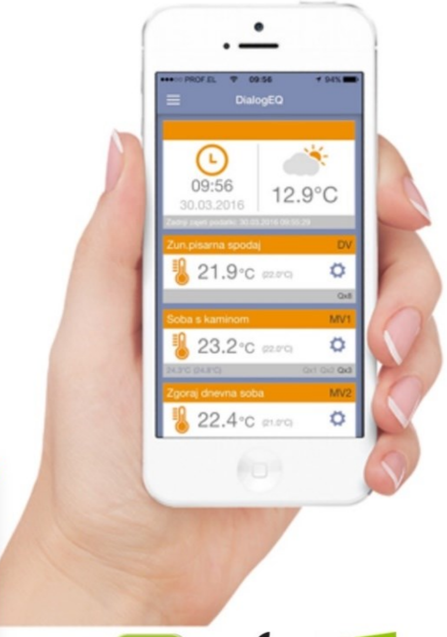
Pritličje
20,4°C
(22,5°C)
30,9°C (33,8) 1 2 3

Nadstropje
20,8°C
(21,5°C)
29,1°C (30,5) 4 5 6

Sanitarna voda
35,5°C
(26,0°C)
22,2°C 9

Glavni vir
33,8°C
(10,0°C)
23,6°C 29,6°C
Jakost
Pretok
Moč (W
Poraba
Delovn

Zunaj
-5,6°C



PROF EL 09:56 94%
DialogEQ

09:56
30.03.2016
12.9°C
Zadnji zajeti podatki: 30.03.2016 09:55:29

Zun.pisarna spodaj DV
21.9°C (22.0°C)
Qx8

Soba s kaminom MV1
23.2°C (22.0°C)
24.3°C (24.8°C) Qx1 Qx2 Qx3

Zgoraj dnevna soba MV2
22.4°C (21.0°C)

Android iOS Windows

2. Programiranje vgrajenih sistemov - Primeri

C. DIALOG EQ („pametni regulator“)

RTOS – ločeni procesi (REG,TCP,WEB,MODBUS,CANBUS) – zahtevnejša izvedba

Regulator DIALOG EQ (krajše: DEQ) predstavlja najnovejšo generacijo pametnih regulatorjev. **Je v c slovenskega znanja.** V podjetju PROF.EL smo ga razvili, saj želje strank in pa predvsem sodobni tre narekujejo daljinski nadzor in avtomatizacijo ogrevalno/hladilnih sistemov oziroma celovite rešitve za dom (smart house).

Regulator DEQ vam omogoča:

- 24h spremljanje delovanja in upravljanje s sistemom na daljavo (računalnik, tablica, pametni telefon),
- varno shranjevanje vseh uporabnikovih nastavitev v oblaku,
- pregled in analizo delovanja sistema,
- prijazen uporabniški vmesnik (interni WEB, WEB aplikacija, aplikacija za mobilne telefone Android, iOS, Windows),
- beleženje in shranjevanje podatkov (črna skrinjica v oblaku),
- daljinsko pomoč servisne ekipe,
- daljinsko posodabljanje (up-grade) programske opreme.

Algoritmi so pripravljeni za vodenje in nadzor kotlov na biomaso in olje, toplotnih črpalk, sončnih kolektorjev, sanitarne vode s cirkulacijsko črpalko idr.

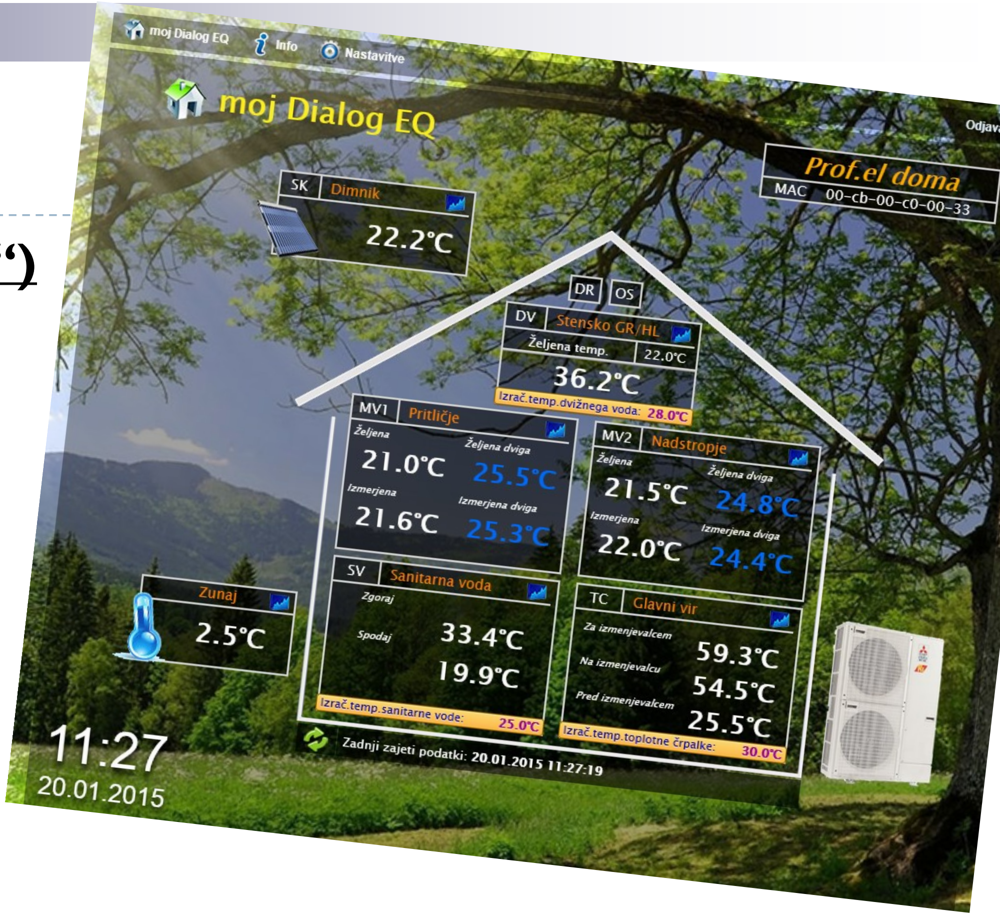
2. Programiranje vgrajenih sistemov - Primeri

C. DIALOG EQ („pametni regulator“)

D13 („HVAC regulator“)

Izhodi in algoritmi za krmiljenje:

- Direktne veje
- Mešalne veje 2x
- Sanitarne vode
- Sončnih kolektorjev



Prof.el doma
MAC 00-cb-00-c0-00-33

SK Dimnik 22.2°C

DR OS

DV Stensko GR/HL
Željena temp. 22.0°C
36.2°C
Izrač.temp.dvižnega voda: 28.0°C

MV1 Pritličje
Željena 21.0°C Željena dviga 25.5°C
Izmerjena 21.6°C Izmerjena dviga 25.3°C

MV2 Nadstropje
Željena 21.5°C Željena dviga 24.8°C
Izmerjena 22.0°C Izmerjena dviga 24.4°C

Zunaj 2.5°C

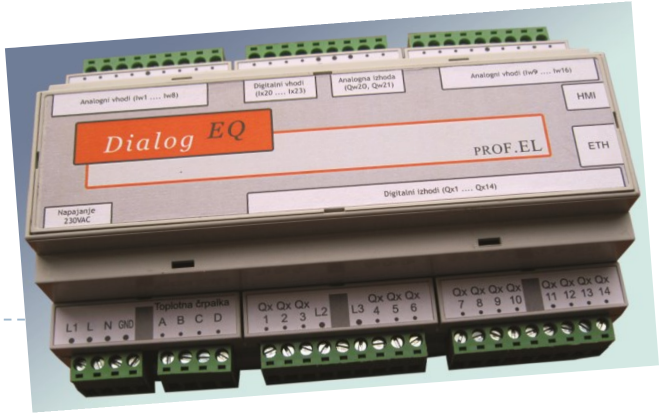
SV Sanitarna voda
Zgoraj 33.4°C
Spodaj 19.9°C
Izrač.temp.sanitarne vode: 25.0°C

TC Glavni vir
Za izmenjevalcem 59.3°C
Na izmenjevalcu 54.5°C
Pred izmenjevalcem 25.5°C
Izrač.temp.toplotne črpalke: 30.0°C

11:27
20.01.2015
Zadnji zajeti podatki: 20.01.2015 11:27:19

Kompleksnejša izvedba:

- MQX RTOS
- Opravila :
  - FP_TASK glavni krmilni program
  - MODBUS_TASK Modbus strežnik
  - TCPCLIENT_TASK povezava s podatkovnim strežnikom v oblaku
  - httpd_server spletni strežnik – lokalni portal
  - CMDEXECUTE_TASK izvedba ukazov
  - FTPCLIENT_TASK FTP prenosi


Analogni vhodi (Iw1 .... Iw8)
Digitalni vhodi (Ix20 .... Ix23)
Analogna izhoda (Qw20, Qw21)
Analogni vhodi (Iw9 .... Iw16)
HMI
Dialog EQ
PROF.EL
ETH
Digitalni izhodi (Qx1 .... Qx14)
Napajanje 230VAC
Toplotna črpalka
L1 L N GND
A B C D
Qx Qx Qx 1 2 3
L2
L3
Qx Qx Qx 4 5 6
Qx Qx Qx Qx 7 8 9 10
Qx Qx Qx Qx 11 12 13 14

2. Programiranje vgrajenih sistemov - Primeri

C. DIALOG EQ („pametni regulator“)

RTOS (primer MQX) :

Opravila („Tasks“)

```
 const TASK_TEMPLATE_STRUCT  MQX_template_list[] =
{
   /* Task Index,      Function,          Stack, Priority,                    Name,                       Attributes,              Param,   Time Slice */
{ MNG_TASK,       MngTask,          1200,  TASK_PRIORITY_MNG_TASK,   MNG_TASK_DES,             MQX_AUTO_START_TASK,       0,          0 },

{ SHELL_TASK,   ShellTask,          2000,  TASK_PRIORITY_SHELL,  SHELL_TASK_DES,     0,                              0,          0 },

{ FP_TASK,          FunPgmTask,       2000,  TASK_PRIORITY_FP,            FP_TASK_DES,                 0,                              0,          0 },

{ TNSH_TASK,      TelnetClientShell,  2000,  TASK_PRIORITY_TNETSH,TNSH_TASK_DES,     0,                              0,          0 },

{ TCPCLIENT_TASK,TCPClient_Task, 2000,  TASK_PRIORITY_TCPCLIENT,TCPCLIENT_TASK_DES,   0,                              0,          0 },

{ MODBUS_TASK,Modbus_Task,    2000,  TASK_PRIORITY_MODBUS,MODBUS_TASK_DES,  0,                              0,          0 },

{ EVTALM_TASK,EventAlmTask,       2000,  TASK_PRIORITY_EVTALM,EVTALM_TASK_DES, 0,                              0,          0 },

{ AIN_TASK,         AinTask,             500,    TASK_PRIORITY_AIN,           AIN_TASK_DES,                0,                              0,          0 },

{ NETMNG_TASK,NetMngTask,        1000,  TASK_PRIORITY_NETMNG,NETMNG_TASK_DES,        0,                              0,          0 },

{ 0 }

};
```

2. Programiranje vgrajenih sistemov - Primeri

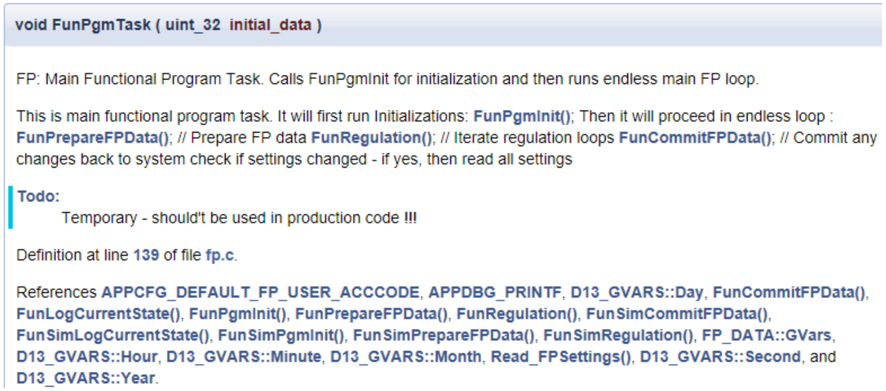
C. DIALOG EQ („pametni regulator“)

RTOS (primer MQX opravila) :

Glavna regulacijska zanka („FP_TASK“)

```
void FunPgmTask (uint_32 initial_data)
{
FunPgmInit();

// register task for system messages
rc = SysMsgRegister ();

// WDT control
WdtRegister (15000, WDT_ACTION_LOG);

// ------------------------------------- main execution loop -----------------------------
while (TRUE)  {

        _time_get_elapsed (&fp_start_time);   //Measure processing time fp_start_time

        WdtReset ();

        FunPrepareFPData();          // Prepare FP data
        FunRegulation();             // Iterate regulation loops
        FunCommitFPData();           // Commit any changes back to system

        _time_get_elapsed (&fp_end_time);   //Measure processing time
        _time_diff (&fp_start_time, &fp_end_time, &fp_loop_time);     // get elapsed time
        FPLoopTime=(fp_loop_time.SECONDS * 1000) + fp_loop_time.MILLISECONDS;

        _time_delay(1000-FPLoopTime);                 // wait for 1000 ms - loop time in ms
        }
_task_block();                                        // Shouldn‘t reach this point
}
```

```
/** @brief FP: Main Functional Program Task.
Calls FunPgmInit for initialization and then runs endless main FP loop.
 *
 *  This is main functional program task.
 *  It will first run Initializations: FunPgmInit();
 *  Then it will proceed in endless loop :
 *              FunPrepareFPData();    // Prepare FP data
 *              FunRegulation();       // Iterate regulation loops
 *              FunCommitFPData();     // Commit any changes back to system
 *              check if settings changed - if yes, then read all settings
*/
```



void FunPgmTask (uint_32 initial_data)

FP: Main Functional Program Task. Calls FunPgmInit for initialization and then runs endless main FP loop.

This is main functional program task. It will first run Initializations: **FunPgmInit()**; Then it will proceed in endless loop : **FunPrepareFPData()**; // Prepare FP data **FunRegulation()**; // Iterate regulation loops **FunCommitFPData()**; // Commit any changes back to system check if settings changed - if yes, then read all settings

Todo:
Temporary - should't be used in production code !!!

Definition at line 139 of file fp.c.

References **APPCFG_DEFAULT_FP_USER_ACCCODE**, **APPDBG_PRINTF**, **D13_GVARS::Day**, **FunCommitFPData()**, **FunLogCurrentState()**, **FunPgmInit()**, **FunPrepareFPData()**, **FunRegulation()**, **FunSimCommitFPData()**, **FunSimLogCurrentState()**, **FunSimPgmInit()**, **FunSimPrepareFPData()**, **FunSimRegulation()**, **FP_DATA::GVars**, **D13_GVARS::Hour**, **D13_GVARS::Minute**, **D13_GVARS::Month**, **Read_FPSettings()**, **D13_GVARS::Second**, and **D13_GVARS::Year**.

D. Tinia – prijazen dom

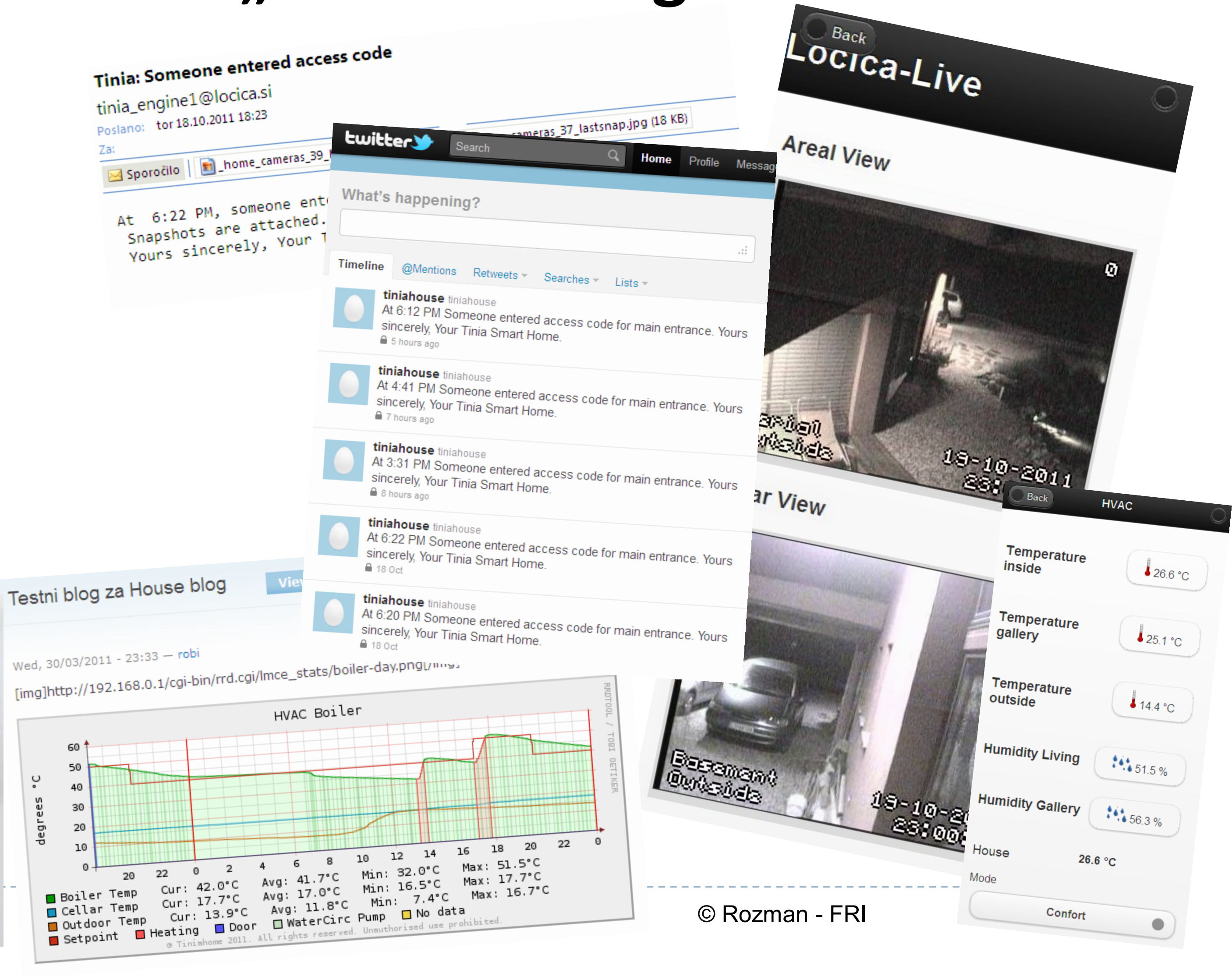
TBS – „Tinia Building Server”

Kratek opis

TBS – „Tinia Building Server”:

Nadzor,upravljanje in vizualizacija delovanja prijaznega doma.

- majhen, varčen, tih (5W)
- povezuje zgradbo in pametno mesto
- informiranje, povratna inf.
  •pametni telefoni, tablice
  •splet, soc.omrežja
- programiranje s pravili,vtičniki
- povezava s soc.omrežji
  •Twitter,FaceBook



Tinia: Someone entered access code

tinia_engine1@locica.si

Poslano: tor 18.10.2011 18:23

Za:

Sporočilo | _home_cameras_39_ | ameras_37_lastsnap.jpg (18 KB)

```
At  6:22 PM, someone ent
 Snapshots are attached.
 Yours sincerely, Your T
```

tiniahouse tiniahouse
At 6:12 PM Someone entered access code for main entrance. Yours sincerely, Your Tinia Smart Home.
5 hours ago

tiniahouse tiniahouse
At 4:41 PM Someone entered access code for main entrance. Yours sincerely, Your Tinia Smart Home.
7 hours ago

tiniahouse tiniahouse
At 3:31 PM Someone entered access code for main entrance. Yours sincerely, Your Tinia Smart Home.
8 hours ago

tiniahouse tiniahouse
At 6:22 PM Someone entered access code for main entrance. Yours sincerely, Your Tinia Smart Home.
18 Oct

tiniahouse tiniahouse
At 6:20 PM Someone entered access code for main entrance. Yours sincerely, Your Tinia Smart Home.
18 Oct

Testni blog za House blog

Wed, 30/03/2011 - 23:33 — robi

[img]http://192.168.0.1/cgi-bin/rrd.cgi/lmce_stats/boiler-day.png[/img]

```
■ Boiler Temp     Cur: 42.0°C   Avg: 41.7°C   Min: 32.0°C   Max: 51.5°C
■ Cellar Temp     Cur: 17.7°C   Avg: 17.0°C   Min: 16.5°C   Max: 17.7°C
■ Outdoor Temp     Cur: 13.9°C   Avg: 11.8°C   Min:  7.4°C   Max: 16.7°C
■ Setpoint   ■ Heating   ■ Door   □ WaterCirc Pump   ■ No data
    © Tiniahome 2011. All rights reserved. Unauthorised use prohibited.
```

Temperature inside	26.6 °C
Temperature gallery	25.1 °C
Temperature outside	14.4 °C
Humidity Living	51.5 %
Humidity Gallery	56.3 %
House	26.6 °C

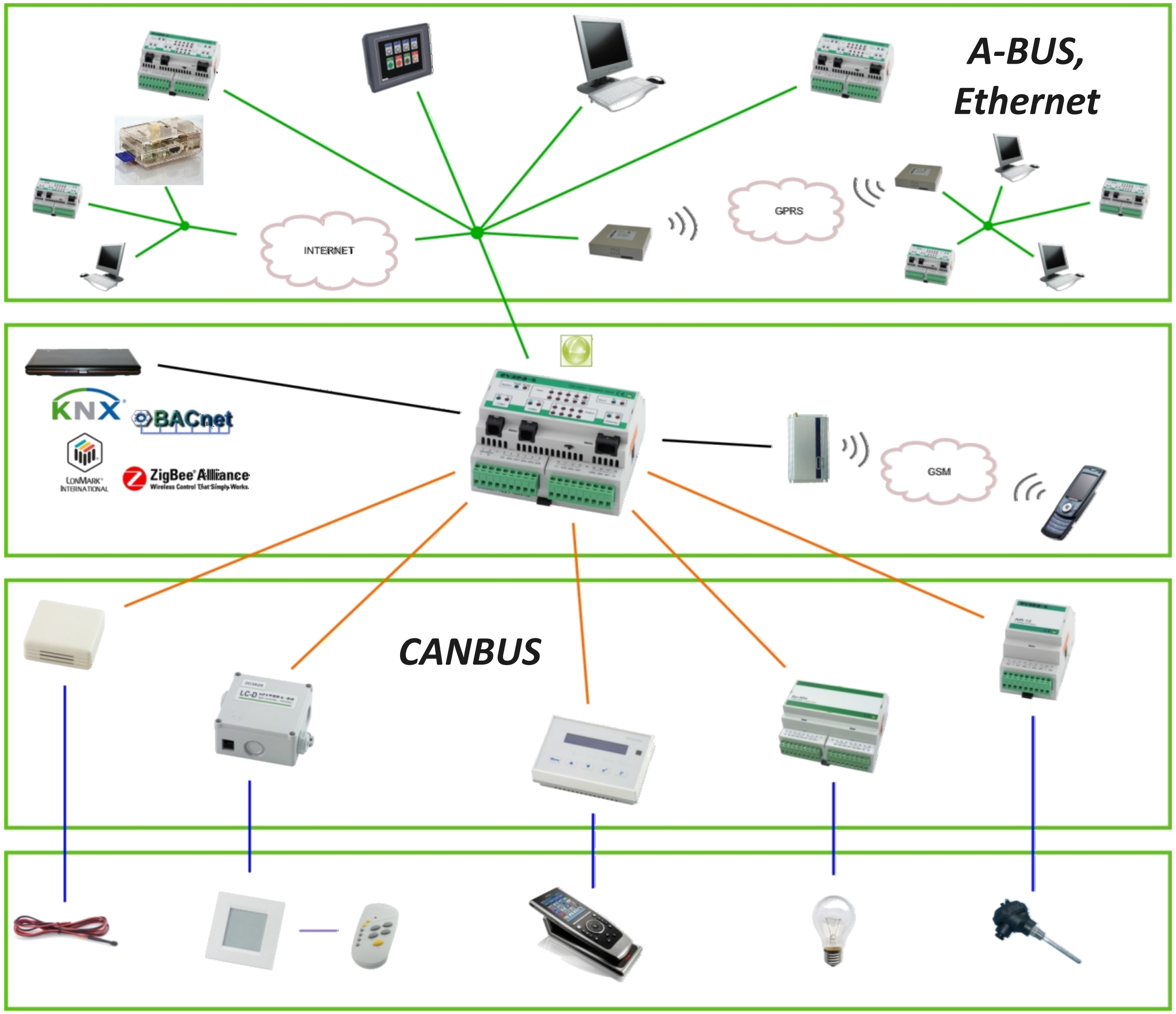
Mode

Confort

INTEGRA BM SYSTEM

Industrial & Building Automation

Generally



High level network

CyBro controller

Low level network

Accessories

Tinia – prijazen dom

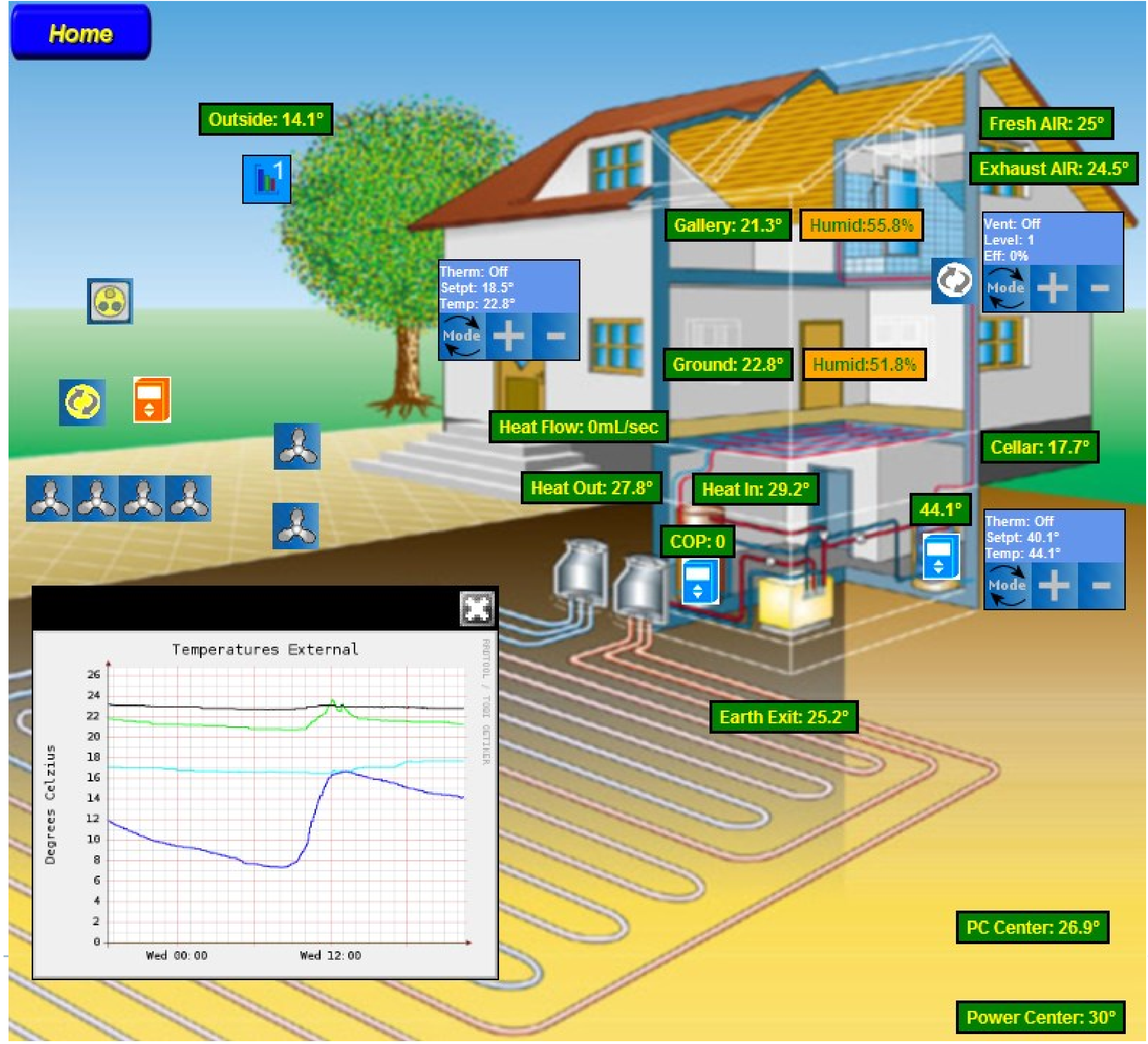
TBS – „Tinia Building Server”

Kratek opis

TBS – „Tinia Building Server”:

***Nadzor,upravljanje in vizualizacija delovanja prijaznega doma*.**

- majhen, varčen, tih (5W)
- povezuje zgradbo in pametno mesto
- informiranje, povratna inf.
  - •pametni telefoni, tablice
  - •splet, soc.omrežja
- programiranje s pravili,vtičniki
- povezava s soc.omrežji
  - •Twitter,FaceBook



Outside: 14.1°

Fresh AIR: 25°

Exhaust AIR: 24.5°

Gallery: 21.3°

Humid:55.8%

Vent: Off
Level: 1
Eff: 0%

Therm: Off
Setpt: 18.5°
Temp: 22.8°

Ground: 22.8°

Humid:51.8%

Heat Flow: 0mL/sec

Cellar: 17.7°

Heat Out: 27.8°

Heat In: 29.2°

44.1°

Therm: Off
Setpt: 40.1°
Temp: 44.1°

COP: 0

Earth Exit: 25.2°

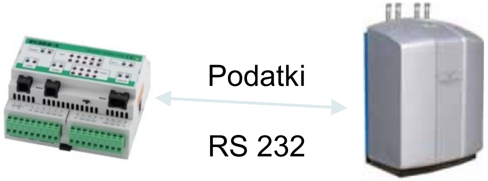
PC Center: 26.9°

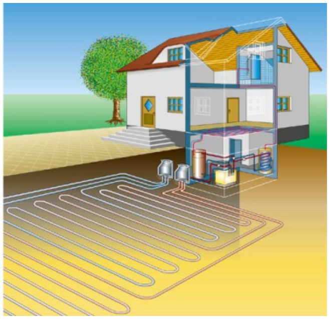
Power Center: 30°

Ogrevanje (prostori, sanitarna voda)

Toplotna črpalka zemlja-voda

- Zemeljski kolektor
- Talno in stensko ogrevanje
- Sanitarna voda
- Serijska komunikacija: Cybro COM2 <-> TČ





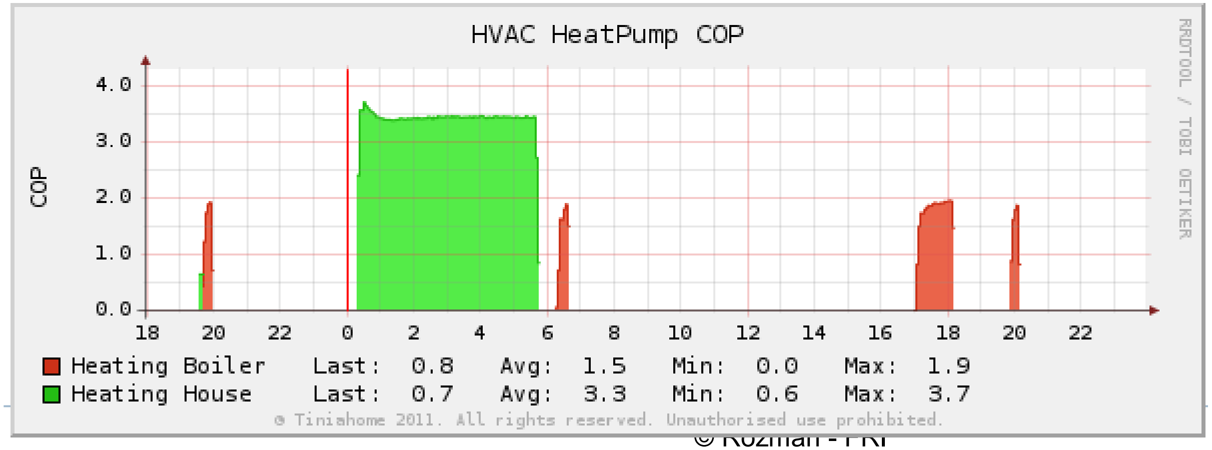
Grelno število (COP-Coefficient Of Performance):

$$COP = \frac{\text{Toplotna Moč}}{\text{Elektricna Moč}}$$

🟩 ▪COP ~ 3.5 Ogrevanje prostorov (Elekt.Moč =1.8 kW)

🟧 ▪COP ~ 2.0 Ogrevanje sanitarne vode (Elekt.Moč =3.0 kW)

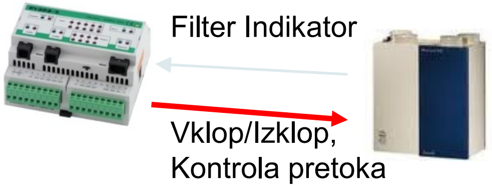
Primer zimskega dneva – COP toplotne črpalke :

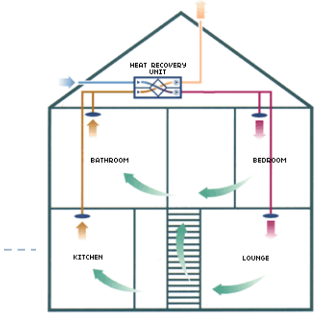


Prezračevanje

Prezračevanje s povratkom toplote (rekuperacija)

- Izčrpan zrak odda toploto svežemu zraku s >85% izkoristkom
- Vklop/izklop in kontrola pretoka
- Indikator zasičenosti filtra



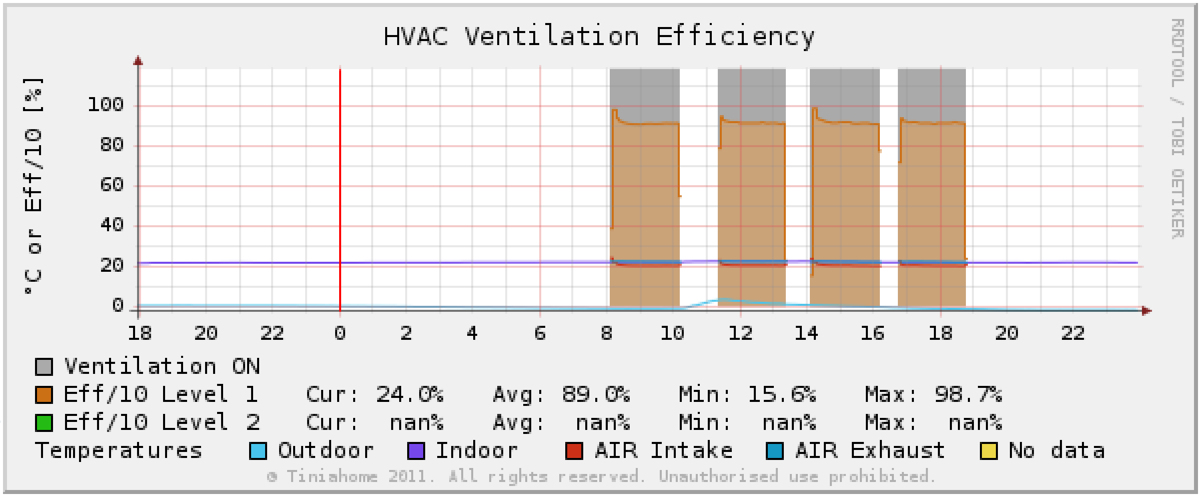


Učinkovitost rekuperacije:

$$\text{EFF.} \approx \frac{\text{Svež zrak temp.} - \text{Zunanji zrak temp.}}{\text{Izčzčrpazrak temp.} - \text{Zunanji zrak temp.}} \ [\%]$$

- Primer : Eff ~ 90% ko:
  - Zunanja Temp. ~0°C
  - Notranja Temp. ~21°C
  - Sveži zrak se segreje od ~0°C do ~19°C (rekuperacija)

Primer zimskega dneva – učinkovitost rekuperacije



Zalivanje (vrt, zelenica, ribnik)

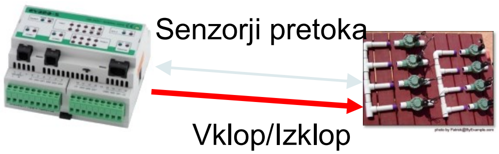
Zalivanje
(vrt, zelenica, ribnik)

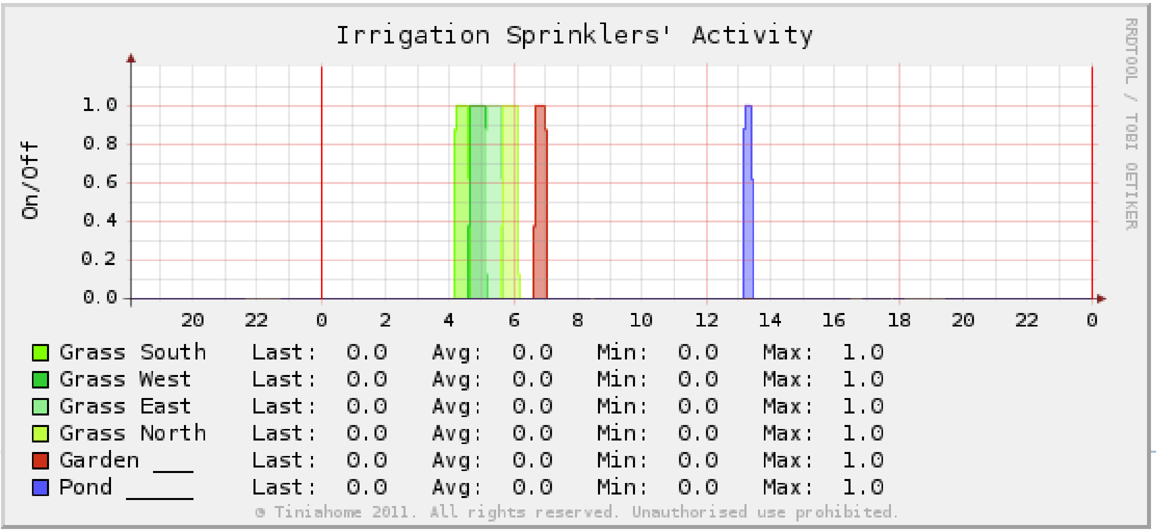
- Kontrola zalivanja v skladu z **urnikom** in **nivojem vlage** v tleh

- Vklop/Izklop & Zaznavanje pretoka

- Zaznavanje pretoka:
  - Preklop med dvema viroma :
    - podtalnica
    - vodovod

Primer poletnega dne - Zalivanje





🟩 Grass South	Last:	0.0	Avg:	0.0	Min:	0.0	Max:	1.0
🟩 Grass West	Last:	0.0	Avg:	0.0	Min:	0.0	Max:	1.0
🟩 Grass East	Last:	0.0	Avg:	0.0	Min:	0.0	Max:	1.0
🟩 Grass North	Last:	0.0	Avg:	0.0	Min:	0.0	Max:	1.0
🟥 Garden ___	Last:	0.0	Avg:	0.0	Min:	0.0	Max:	1.0
🟦 Pond _____	Last:	0.0	Avg:	0.0	Min:	0.0	Max:	1.0

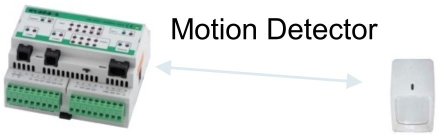
Vzorci gibanja - prisotnost

Detekcija gibanja/prisotnosti

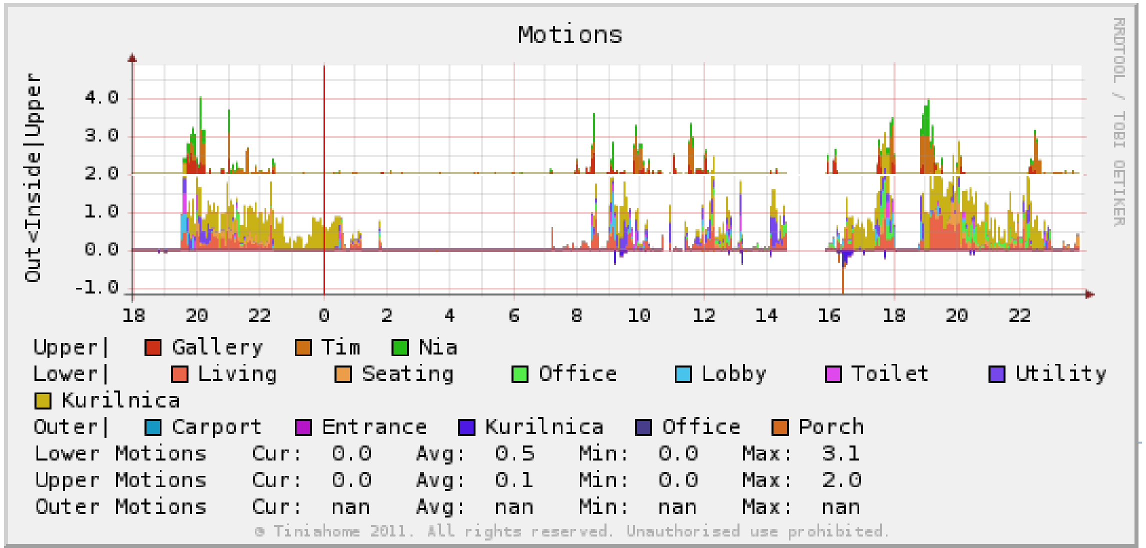
- Detekcija gibanja v posameznih prostorih
- Informacija o prisotnosti

Uporaba vzorcev gibanja oziroma prisotnosti :

- Upravljanje :
  - Razsvetljave
  - Ogrevanja, hlajenja
  - A/V naprav
- Profiliranje, napovedi :
  - Dogodkov v prihodnosti
  - Energetskih potreb
  - Nastavitev



IR beams/PIR motion detector/Dual+MW/roof PIR
PET immune/Curtain PIR with direction...professional

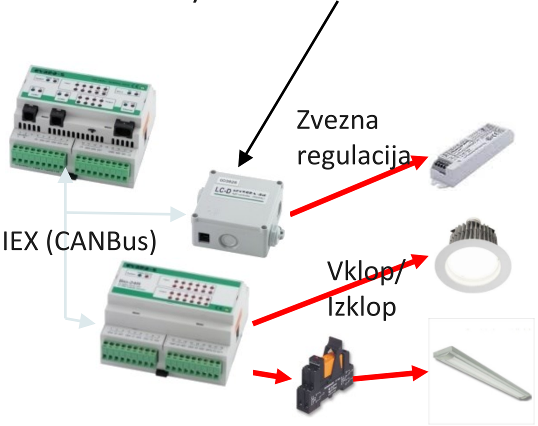


Razsvetljava

Vklop/Izklop in zvezna regulacija razsvetljave

•**Vklop/Izklop** kontrola s Cybrotech moduli BIO-24R in BIO-24T, Zunanjimi releji

•**Zvezna regulacija** s pomočjo DALI balastov - Cybrotech modul LC-D



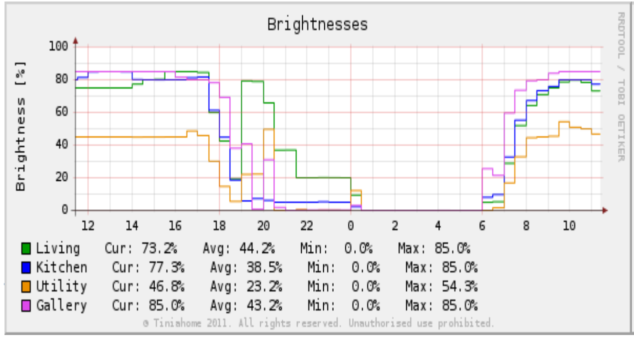
- Luči se upravljajo v skupinah

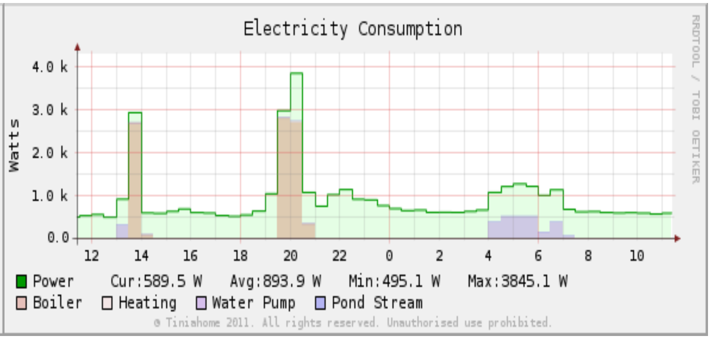
- Običajno krmiljena s pomočjo scen in zaznavanje osvetljenosti:
  - **Statične scene** – npr. : Prehrana,Obisk, Romantika,TV, Branje,Relaksacija, …
  - **Dogodkovne scene**: Ko se vklopi TV, nastavi bližnjo luč na 20%.

- Zmanjševanje porabe :
  - **Časovne luči** (izklopi po določenem času odsotnosti)
  - Vklopi luč samo, ko je to **res potrebno** (trenutna osvetljenost)
  - Nastavi zvezne luči samo na **potrebno stopnjo** (glede na osvetljenost)

Primer meritev osvetljenosti in nadzora porabe el. energije

(glavni porabnik el. Energije so posebej izpostavljeni)





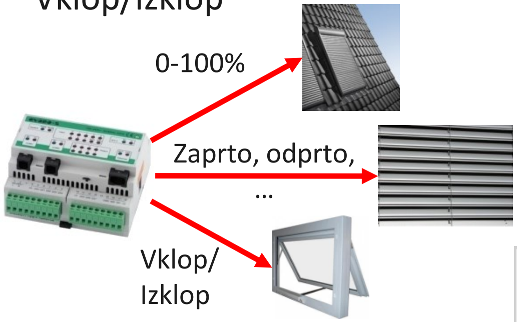
Pasivno ogrevanje/hlajenje…

Rolete, Žaluzije, Okna

•**Rolete**: med 0% - 100%
(0% odprte, 100% zaprte)

•**Žaluzije** imajo stanja :
Zaprto(100%), Senčeno(75%), Odprto(50%), Solarno pasivno (25%), Dvignjeno(0%).

•**Motorizirana** okna:
Vklop/Izklop



▪Strešna okna z roletami :

- ▪Severna, običajno:
  - ▪**Odprta v toplem vremenu** za boljšo osvetlitev
  - ▪**Zaprta v hladnem vremenu** za ohranjanje toplote
- ▪Južna, običajno:
  - ▪**Odprta v hladnem, sončnem vremenu** za pasivno ogrevanje
  - ▪**Zaprta v vročem vremenu** proti pregrevanju

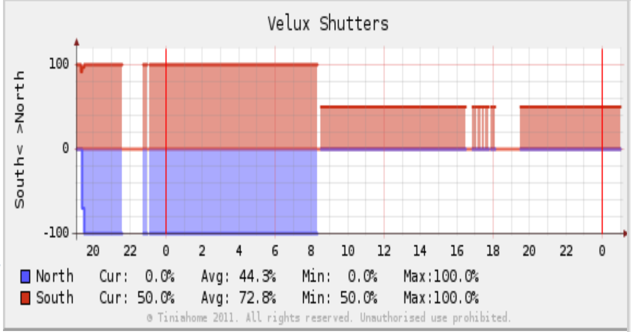
▪Žaluzije:

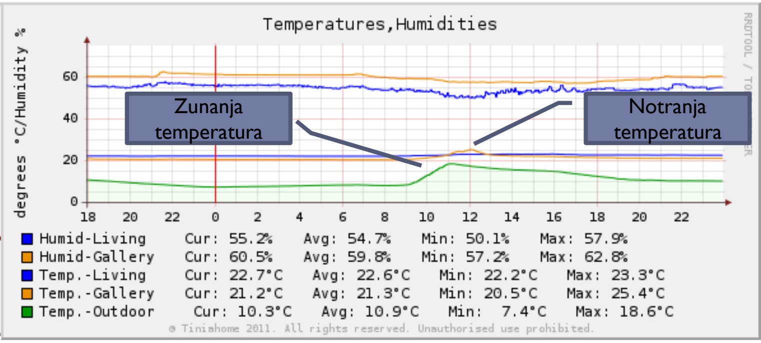
- ▪**Senčene ali zaprte ob izrazitem sončnem vremenu poleti**
- ▪**Odprte v “solarni” poziciji ob sončnih dnevih pozimi**

▪Motorizirana okna (s komarniki) :

- ▪**Odprta v poletnih nočeh za pasivno ohlajanje**

Primer stanj rolet in temperatur v sončnem zimskem dnevu:

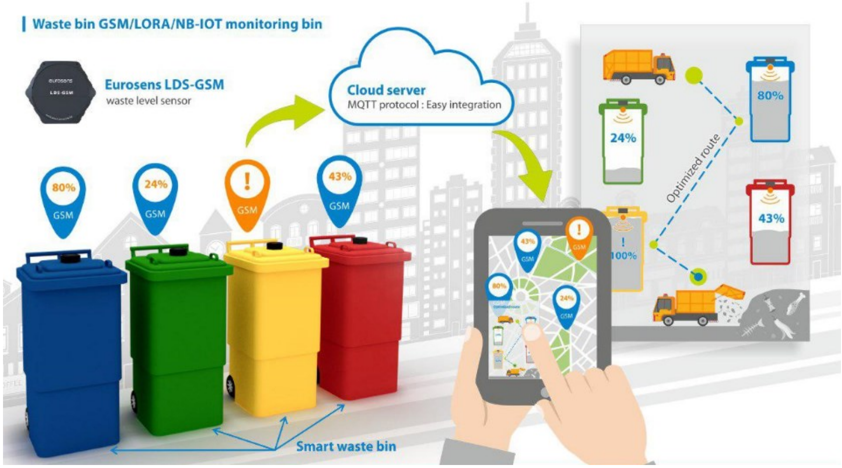


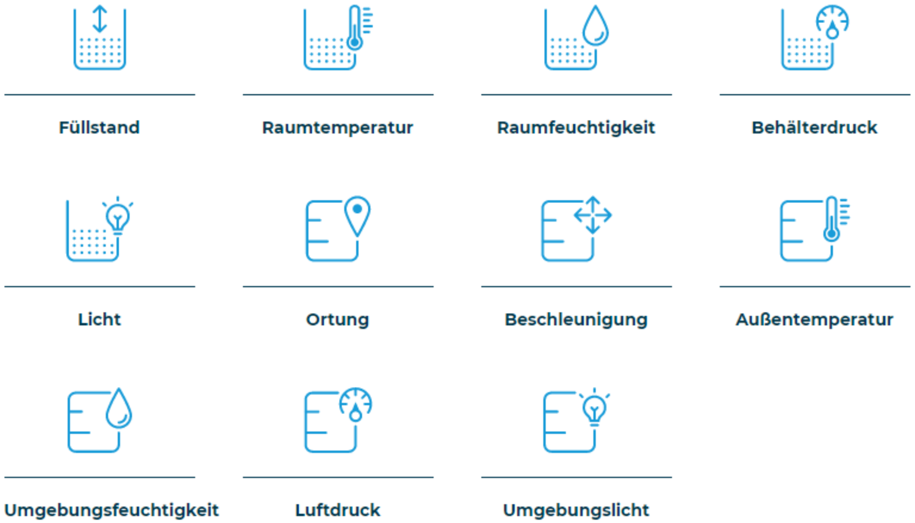


2. Programiranje vgrajenih sistemov - Primeri

E. Pametni zabojnik

Merilnik tlaka, temperature, nivoja, pozicije, … za pametne zabojnike

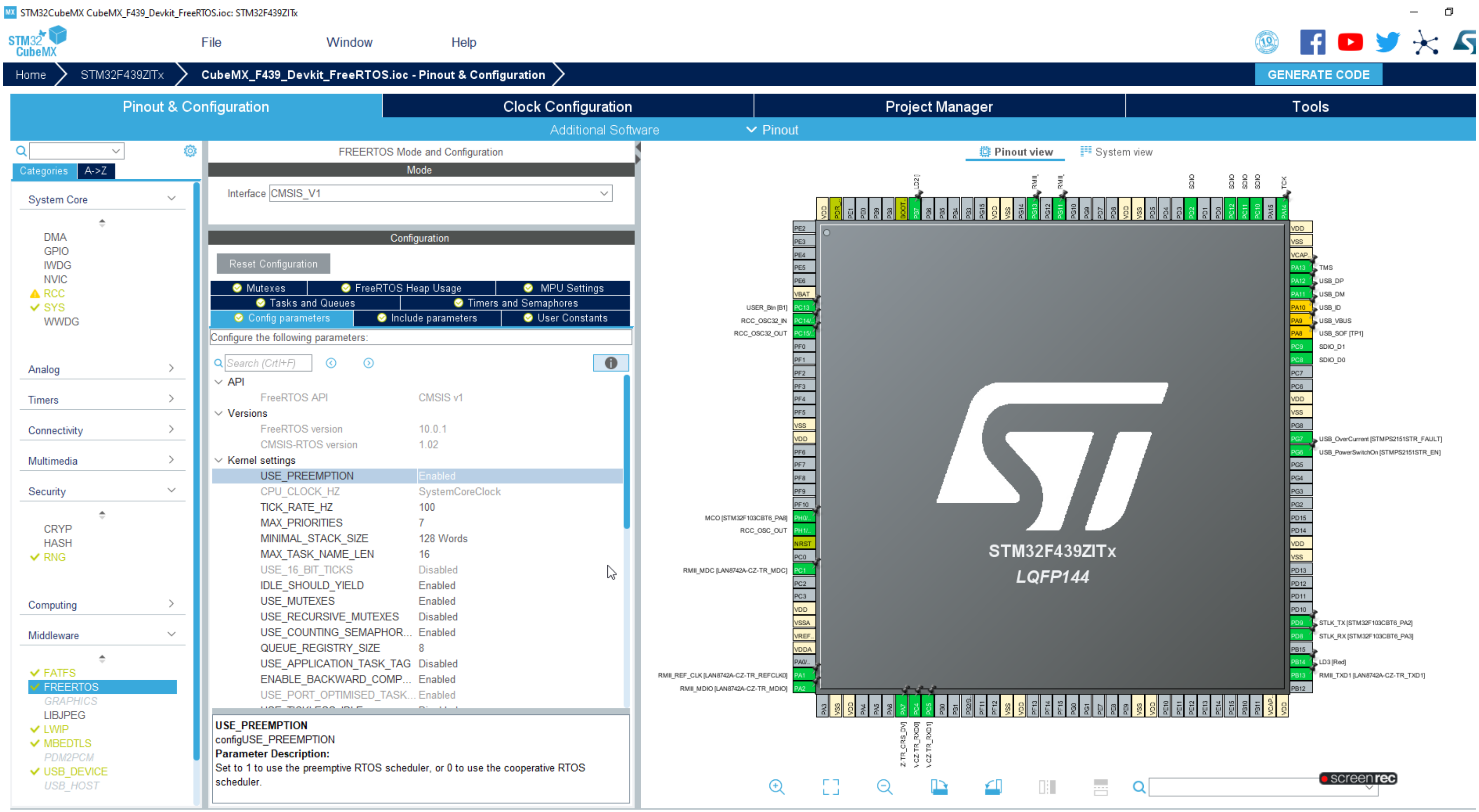




2. Programiranje vgrajenih sistemov - Primeri

E. Pametni zabojnik

Merilnik tlaka, temperature, nivoja, pozicije, … za pametne zabojnike



Interface CMSIS_V1

Configure the following parameters:

API
- FreeRTOS API: CMSIS v1

Versions
- FreeRTOS version: 10.0.1
- CMSIS-RTOS version: 1.02

Kernel settings

USE_PREEMPTION	Enabled
CPU_CLOCK_HZ	SystemCoreClock
TICK_RATE_HZ	100
MAX_PRIORITIES	7
MINIMAL_STACK_SIZE	128 Words
MAX_TASK_NAME_LEN	16
USE_16_BIT_TICKS	Disabled
IDLE_SHOULD_YIELD	Enabled
USE_MUTEXES	Enabled
USE_RECURSIVE_MUTEXES	Disabled
USE_COUNTING_SEMAPHOR...	Enabled
QUEUE_REGISTRY_SIZE	8
USE_APPLICATION_TASK_TAG	Disabled
ENABLE_BACKWARD_COMP...	Enabled
USE_PORT_OPTIMISED_TASK...	Enabled

USE_PREEMPTION
configUSE_PREEMPTION
Parameter Description:
Set to 1 to use the preemptive RTOS scheduler, or 0 to use the cooperative RTOS scheduler.

2. Programiranje vgrajenih sistemov - Primeri

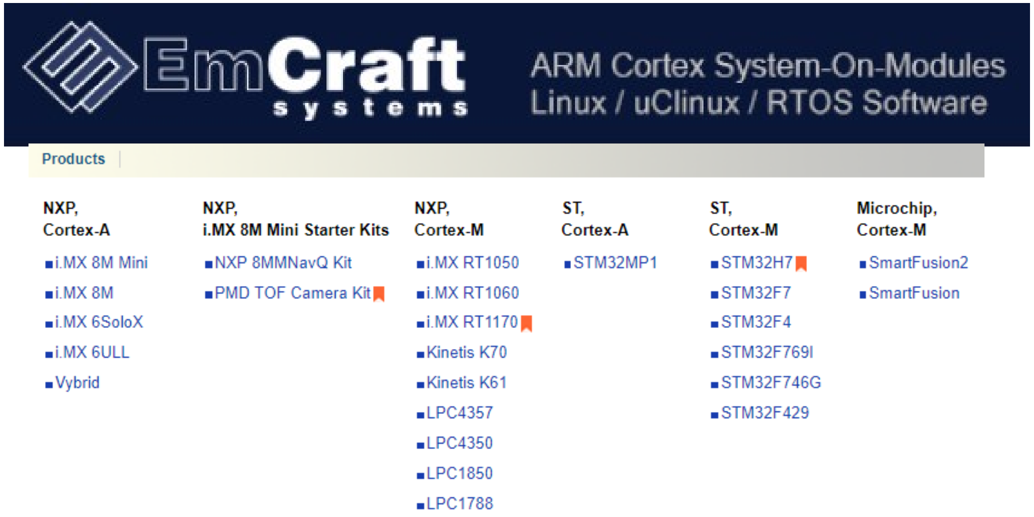
F. Embedded Linux (UcLinux, Buildroot)

Buildroot na STM32F769

bootlin

- Understanding the Linux graphics stack
- Real-time Linux with PREEMPT_RT

https://bootlin.com/



EmCraft systems

ARM Cortex System-On-Modules
Linux / uClinux / RTOS Software

Products

NXP, Cortex-A
- i.MX 8M Mini
- i.MX 8M
- i.MX 6SoloX
- i.MX 6ULL
- Vybrid

NXP, i.MX 8M Mini Starter Kits
- NXP 8MMNavQ Kit
- PMD TOF Camera Kit

NXP, Cortex-M
- i.MX RT1050
- i.MX RT1060
- i.MX RT1170
- Kinetis K70
- Kinetis K61
- LPC4357
- LPC4350
- LPC1850
- LPC1788

ST, Cortex-A
- STM32MP1

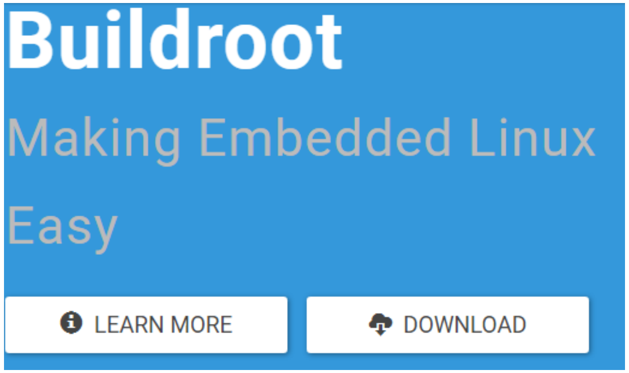
ST, Cortex-M
- STM32H7
- STM32F7
- STM32F4
- STM32F769I
- STM32F746G
- STM32F429

Microchip, Cortex-M
- SmartFusion2
- SmartFusion

https://www.emcraft.com/

yocto PROJECT

THE YOCTO PROJECT. IT'S NOT AN EMBEDDED LINUX DISTRIBUTION,
IT CREATES A CUSTOM ONE FOR YOU.



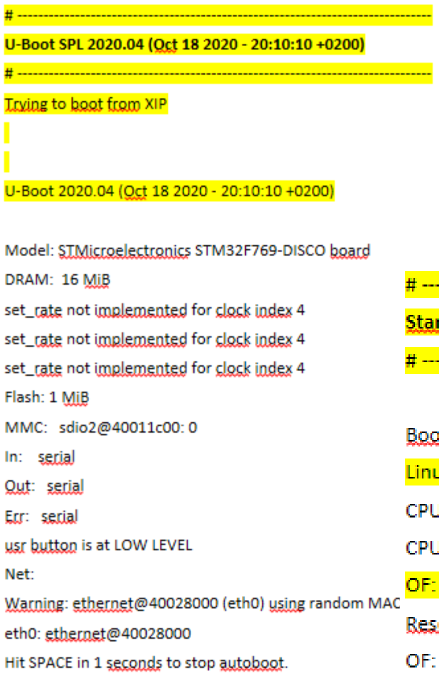
Buildroot

Making Embedded Linux Easy

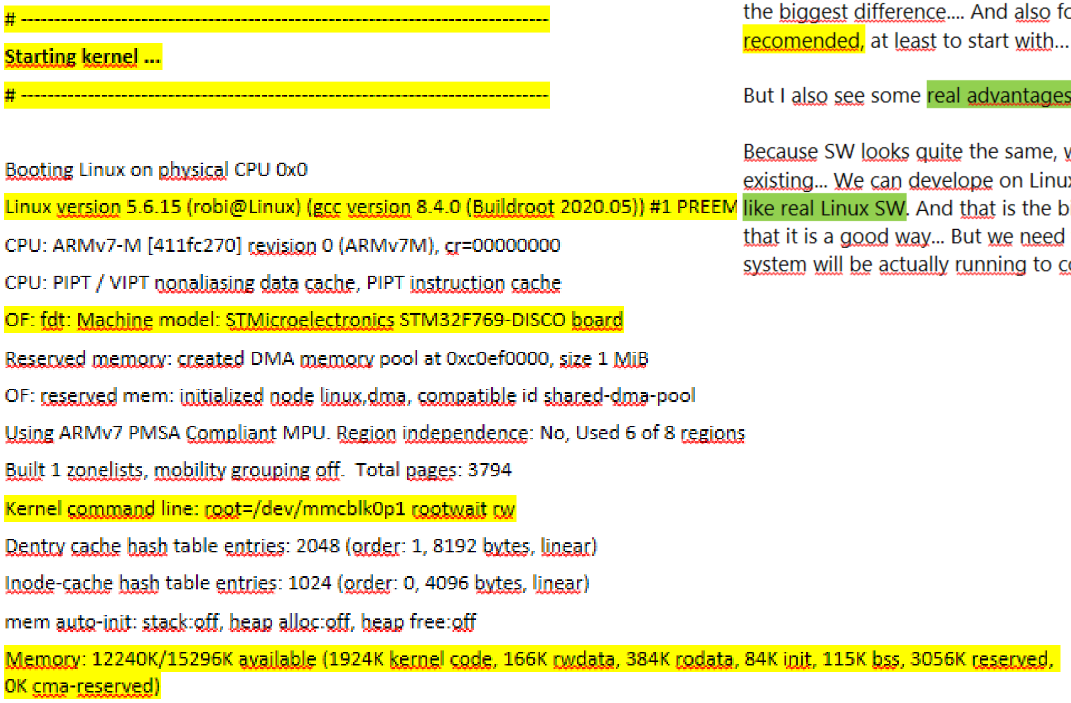
2. Programiranje vgrajenih sistemov - Primeri

F. Embedded Linux (UcLinux, Buildroot)

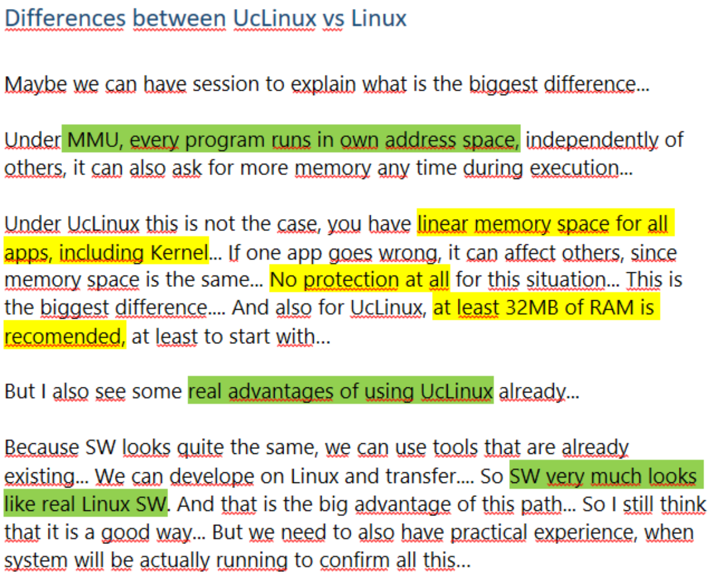
Buildroot na STM32F769



```
# --------------------------------------------------------------------------------
U-Boot SPL 2020.04 (Oct 18 2020 - 20:10:10 +0200)
# --------------------------------------------------------------------------------
Trying to boot from XIP


U-Boot 2020.04 (Oct 18 2020 - 20:10:10 +0200)

Model: STMicroelectronics STM32F769-DISCO board
DRAM:  16 MiB
set_rate not implemented for clock index 4
set_rate not implemented for clock index 4
set_rate not implemented for clock index 4
Flash: 1 MiB
MMC:   sdio2@40011c00: 0
In:    serial
Out:   serial
Err:   serial
usr button is at LOW LEVEL
Net:
Warning: ethernet@40028000 (eth0) using random MAC
eth0: ethernet@40028000
Hit SPACE in 1 seconds to stop autoboot.
```



```
# --------------------------------------------------------------------------------
Starting kernel ...
# --------------------------------------------------------------------------------

Booting Linux on physical CPU 0x0
Linux version 5.6.15 (robi@Linux) (gcc version 8.4.0 (Buildroot 2020.05)) #1 PREEM
CPU: ARMv7-M [411fc270] revision 0 (ARMv7M), cr=00000000
CPU: PIPT / VIPT nonaliasing data cache, PIPT instruction cache
OF: fdt: Machine model: STMicroelectronics STM32F769-DISCO board
Reserved memory: created DMA memory pool at 0xc0ef0000, size 1 MiB
OF: reserved mem: initialized node linux,dma, compatible id shared-dma-pool
Using ARMv7 PMSA Compliant MPU. Region independence: No, Used 6 of 8 regions
Built 1 zonelists, mobility grouping off.  Total pages: 3794
Kernel command line: root=/dev/mmcblk0p1 rootwait rw
Dentry cache hash table entries: 2048 (order: 1, 8192 bytes, linear)
Inode-cache hash table entries: 1024 (order: 0, 4096 bytes, linear)
mem auto-init: stack:off, heap alloc:off, heap free:off
Memory: 12240K/15296K available (1924K kernel code, 166K rwdata, 384K rodata, 84K init, 115K bss, 3056K reserved,
0K cma-reserved)
```



Differences between UcLinux vs Linux

Maybe we can have session to explain what is the biggest difference…

Under MMU, every program runs in own address space, independently of others, it can also ask for more memory any time during execution…

Under UcLinux this is not the case, you have linear memory space for all apps, including Kernel… If one app goes wrong, it can affect others, since memory space is the same… No protection at all for this situation… This is the biggest difference…. And also for UcLinux, at least 32MB of RAM is recomended, at least to start with…

But I also see some real advantages of using UcLinux already…

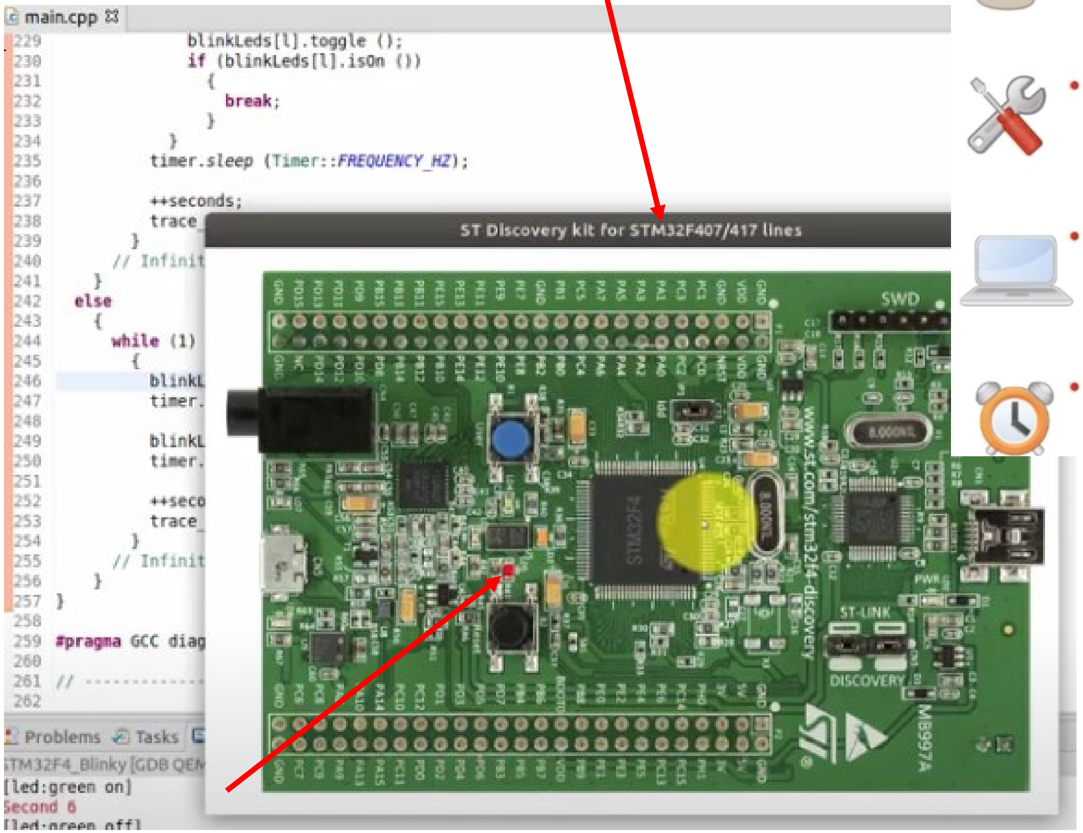
Because SW looks quite the same, we can use tools that are already existing… We can develope on Linux and transfer…. So SW very much looks like real Linux SW. And that is the big advantage of this path… So I still think that it is a good way… But we need to also have practical experience, when system will be actually running to confirm all this…

2. Programiranje vgrajenih sistemov - Primeri

G. Simulacije, Emulacije

QEMU – STM32 Discovery

https://www.qemu.org/



main.cpp

```
            blinkLeds[l].toggle ();
            if (blinkLeds[l].isOn ())
              {
                break;
              }
          }
        timer.sleep (Timer::FREQUENCY_HZ);

        ++seconds;
        trace
      }
    // Infinit
  }
else
  {
    while (1)
      {
        blinkL
        timer.

        blinkL
        timer.

        ++seco
        trace_
      }
    // Infinit
  }
}

#pragma GCC diag

// ............
```

Problems Tasks

```
STM32F4_Blinky [GDB QEM
[led:green on]
Second 6
[led:green off]
```

ST Discovery kit for STM32F407/417 lines
SWD
www.st.com/stm32f4-discovery
PWR
ST-LINK
DISCOVERY
MB997A

Why use QEMU?

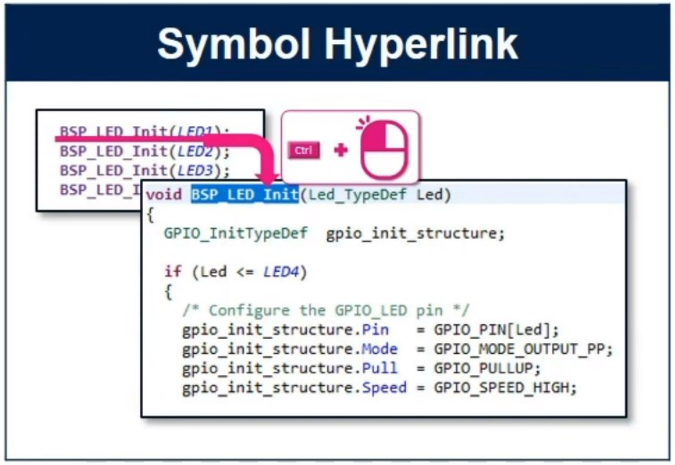
QEMU

A generic and open source machine emulator and virtualizer

- Cost
  - free and open source software (GPLv2)
  - no development kit required
- Experiment without fear
  - Minimize the risk of corrupting valuable development boards
- Portability
  - Not tied to a lab bench --> only need QEMU and a laptop
- Reduce project timescales
  - work in advance of prototype board or silicon delivery

2. Programiranje vgrajenih sistemov

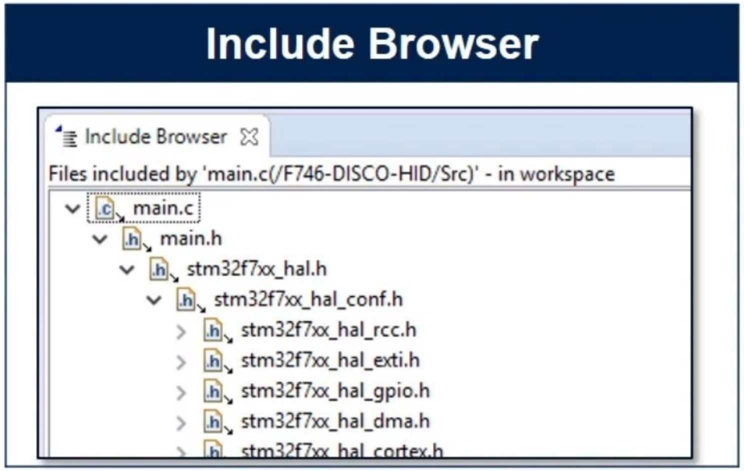
Razvoj in razhroščevanje (primer CubeIDE)



Symbol Hyperlink

```
BSP_LED_Init(LED1);
BSP_LED_Init(LED2);
BSP_LED_Init(LED3);
BSP_LED_I
```

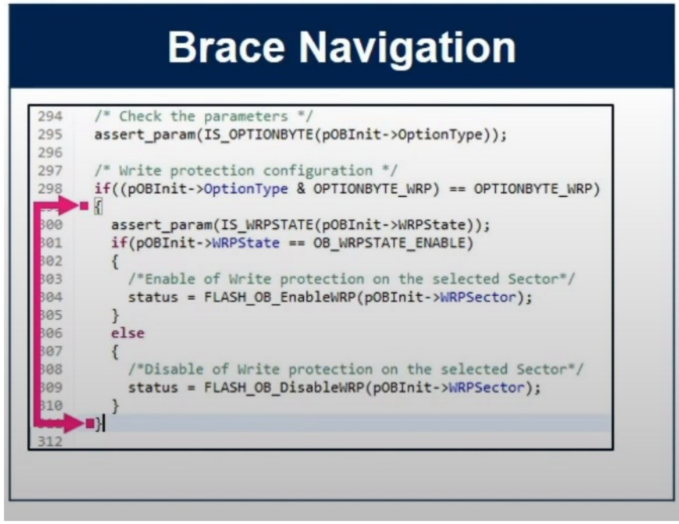
```
void BSP_LED_Init(Led_TypeDef Led)
{
  GPIO_InitTypeDef  gpio_init_structure;

  if (Led <= LED4)
  {
    /* Configure the GPIO_LED pin */
    gpio_init_structure.Pin   = GPIO_PIN[Led];
    gpio_init_structure.Mode  = GPIO_MODE_OUTPUT_PP;
    gpio_init_structure.Pull  = GPIO_PULLUP;
    gpio_init_structure.Speed = GPIO_SPEED_HIGH;
```



Include Browser

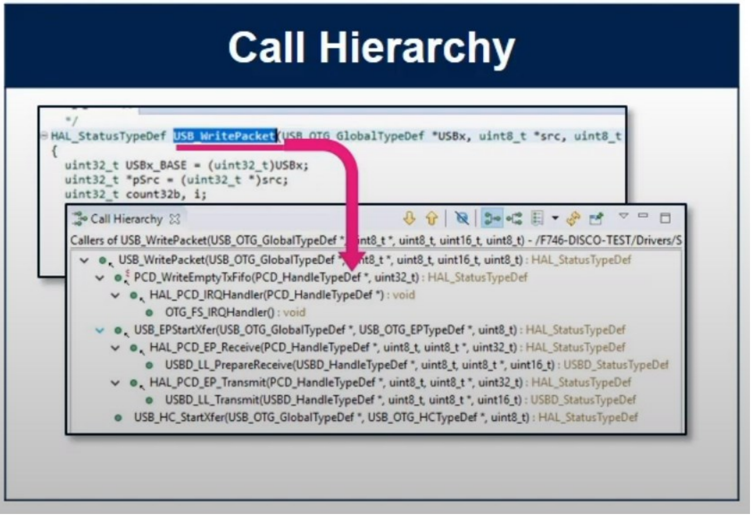
Files included by 'main.c(/F746-DISCO-HID/Src)' - in workspace

```
main.c
  main.h
    stm32f7xx_hal.h
      stm32f7xx_hal_conf.h
        stm32f7xx_hal_rcc.h
        stm32f7xx_hal_exti.h
        stm32f7xx_hal_gpio.h
        stm32f7xx_hal_dma.h
        stm32f7xx_hal_cortex.h
```



Brace Navigation

```
  /* Check the parameters */
  assert_param(IS_OPTIONBYTE(pOBInit->OptionType));

  /* Write protection configuration */
  if((pOBInit->OptionType & OPTIONBYTE_WRP) == OPTIONBYTE_WRP)
  {
    assert_param(IS_WRPSTATE(pOBInit->WRPState));
    if(pOBInit->WRPState == OB_WRPSTATE_ENABLE)
    {
      /*Enable of Write protection on the selected Sector*/
      status = FLASH_OB_EnableWRP(pOBInit->WRPSector);
    }
    else
    {
      /*Disable of Write protection on the selected Sector*/
      status = FLASH_OB_DisableWRP(pOBInit->WRPSector);
    }
  }
```



Call Hierarchy

```
  */
HAL_StatusTypeDef USB_WritePacket(USB_OTG_GlobalTypeDef *USBx, uint8_t *src, uint8_t
{
  uint32_t USBx_BASE = (uint32_t)USBx;
  uint32_t *pSrc = (uint32_t *)src;
  uint32_t count32b, i;
```

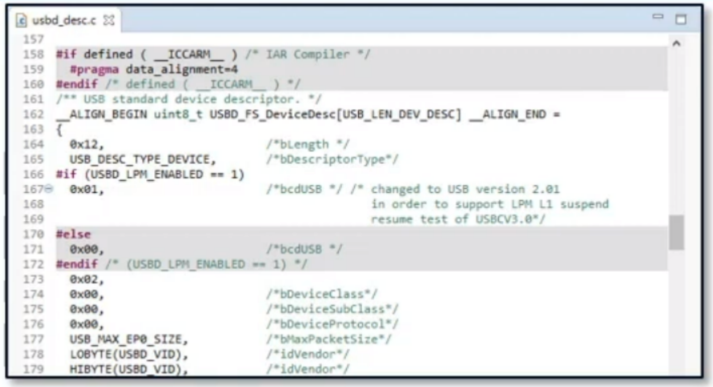
Callers of USB_WritePacket(USB_OTG_GlobalTypeDef *, uint8_t *, uint8_t, uint16_t, uint8_t) - /F746-DISCO-TEST/Drivers/S

```
USB_WritePacket(USB_OTG_GlobalTypeDef *, uint8_t *, uint8_t, uint16_t, uint8_t) : HAL_StatusTypeDef
  PCD_WriteEmptyTxFifo(PCD_HandleTypeDef *, uint32_t) : HAL_StatusTypeDef
    HAL_PCD_IRQHandler(PCD_HandleTypeDef *) : void
      OTG_FS_IRQHandler() : void
  USB_EPStartXfer(USB_OTG_GlobalTypeDef *, USB_OTG_EPTypeDef *, uint8_t) : HAL_StatusTypeDef
    HAL_PCD_EP_Receive(PCD_HandleTypeDef *, uint8_t, uint8_t *, uint32_t) : HAL_StatusTypeDef
      USBD_LL_PrepareReceive(USBD_HandleTypeDef *, uint8_t, uint8_t *, uint16_t) : USBD_StatusTypeDef
    HAL_PCD_EP_Transmit(PCD_HandleTypeDef *, uint8_t, uint8_t *, uint32_t) : HAL_StatusTypeDef
      USBD_LL_Transmit(USBD_HandleTypeDef *, uint8_t, uint8_t *, uint16_t) : USBD_StatusTypeDef
  USB_HC_StartXfer(USB_OTG_GlobalTypeDef *, USB_OTG_HCTypeDef *, uint8_t) : HAL_StatusTypeDef
```

2. Programiranje vgrajenih sistemov

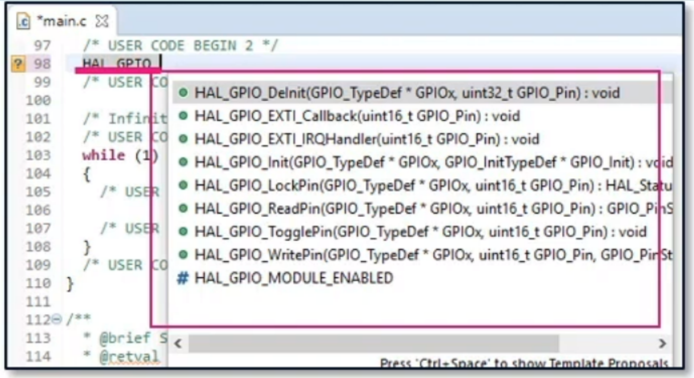
Razvoj in razhroščevanje (primer CubeIDE)

Highlight Inactive Code



```
#if defined ( __ICCARM__ ) /* IAR Compiler */
  #pragma data_alignment=4
#endif /* defined ( __ICCARM__ ) */
/** USB standard device descriptor. */
__ALIGN_BEGIN uint8_t USBD_FS_DeviceDesc[USB_LEN_DEV_DESC] __ALIGN_END =
{
  0x12,                       /*bLength */
  USB_DESC_TYPE_DEVICE,       /*bDescriptorType*/
#if (USBD_LPM_ENABLED == 1)
  0x01,                       /*bcdUSB */ /* changed to USB version 2.01
                                             in order to support LPM L1 suspend
                                             resume test of USBCV3.0*/
#else
  0x00,                       /*bcdUSB */
#endif /* (USBD_LPM_ENABLED == 1) */
  0x02,
  0x00,                       /*bDeviceClass*/
  0x00,                       /*bDeviceSubClass*/
  0x00,                       /*bDeviceProtocol*/
  USB_MAX_EP0_SIZE,           /*bMaxPacketSize*/
  LOBYTE(USBD_VID),           /*idVendor*/
  HIBYTE(USBD_VID),           /*idVendor*/
```

Auto-Complete

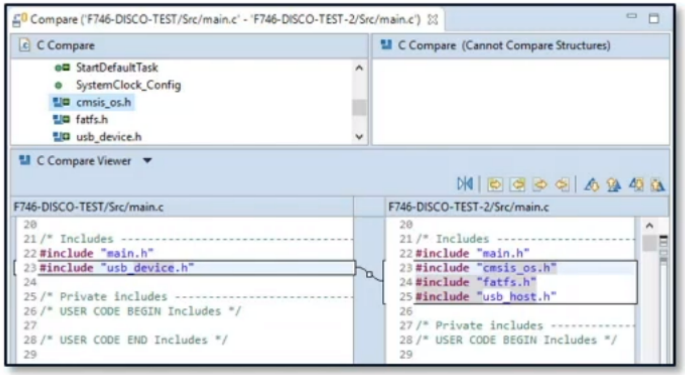


```
  /* USER CODE BEGIN 2 */
  HAL_GPIO
  /* USER CO

  /* Infinit
  /* USER CO
  while (1)
  {
    /* USER

    /* USER
  }
  /* USER CO
}

/**
  * @brief S
  * @retval
```

- HAL_GPIO_DeInit(GPIO_TypeDef * GPIOx, uint32_t GPIO_Pin) : void
- HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) : void
- HAL_GPIO_EXTI_IRQHandler(uint16_t GPIO_Pin) : void
- HAL_GPIO_Init(GPIO_TypeDef * GPIOx, GPIO_InitTypeDef * GPIO_Init) : void
- HAL_GPIO_LockPin(GPIO_TypeDef * GPIOx, uint16_t GPIO_Pin) : HAL_Statu
- HAL_GPIO_ReadPin(GPIO_TypeDef * GPIOx, uint16_t GPIO_Pin) : GPIO_PinS
- HAL_GPIO_TogglePin(GPIO_TypeDef * GPIOx, uint16_t GPIO_Pin) : void
- HAL_GPIO_WritePin(GPIO_TypeDef * GPIOx, uint16_t GPIO_Pin, GPIO_PinSt
- # HAL_GPIO_MODULE_ENABLED

Press 'Ctrl+Space' to show Template Proposals

File Diff/Compare



Compare ('F746-DISCO-TEST/Src/main.c' - 'F746-DISCO-TEST-2/Src/main.c')
C Compare
StartDefaultTask
SystemClock_Config
cmsis_os.h
fatfs.h
usb_device.h
C Compare (Cannot Compare Structures)
C Compare Viewer

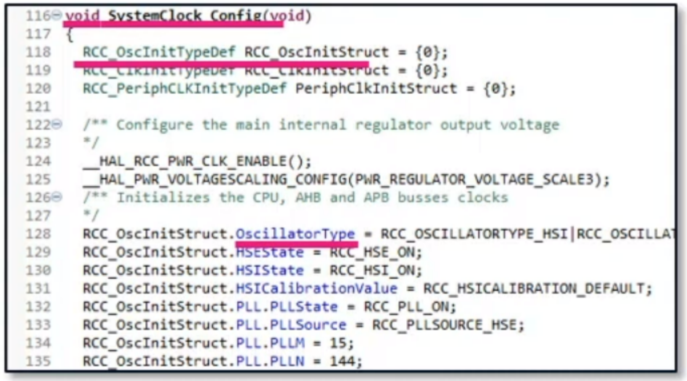
F746-DISCO-TEST/Src/main.c

```

/* Includes -----------------------------------
#include "main.h"
#include "usb_device.h"

/* Private includes ---------------------------
/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

```

F746-DISCO-TEST-2/Src/main.c

```

/* Includes ----------------------
#include "main.h"
#include "cmsis_os.h"
#include "fatfs.h"
#include "usb_host.h"

/* Private includes ------------
/* USER CODE BEGIN Includes */

```

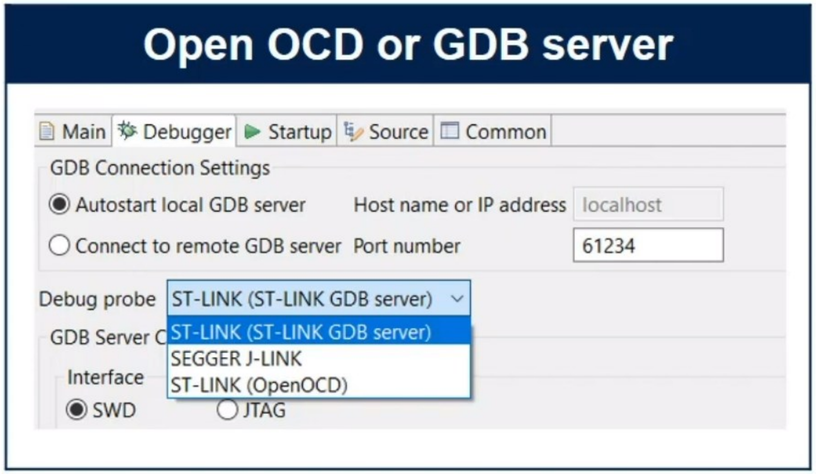
Syntax Highlight



```
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
  RCC_PeriphCLKInitTypeDef PeriphClkInitStruct = {0};

  /** Configure the main internal regulator output voltage
  */
  __HAL_RCC_PWR_CLK_ENABLE();
  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE3);
  /** Initializes the CPU, AHB and APB busses clocks
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI|RCC_OSCILLAT
  RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  RCC_OscInitStruct.HSIState = RCC_HSI_ON;
  RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
  RCC_OscInitStruct.PLL.PLLM = 15;
  RCC_OscInitStruct.PLL.PLLN = 144;
```

2. Programiranje vgrajenih sistemov

Razvoj in razhroščevanje (primer CubeIDE)



Open OCD or GDB server

Main | Debugger | Startup | Source | Common

GDB Connection Settings

● Autostart local GDB server   Host name or IP address: localhost

○ Connect to remote GDB server   Port number: 61234

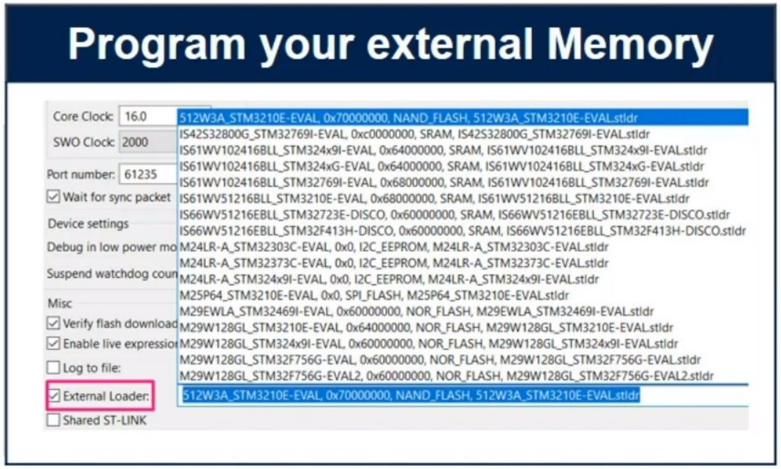
Debug probe: ST-LINK (ST-LINK GDB server)

- ST-LINK (ST-LINK GDB server)
- SEGGER J-LINK
- ST-LINK (OpenOCD)

GDB Server C

Interface

● SWD   ○ JTAG



Program your external Memory

Core Clock: 16.0

SWO Clock: 2000

Port number: 61235

☑ Wait for sync packet

Device settings

Debug in low power mo

Suspend watchdog coun

Misc

☑ Verify flash download

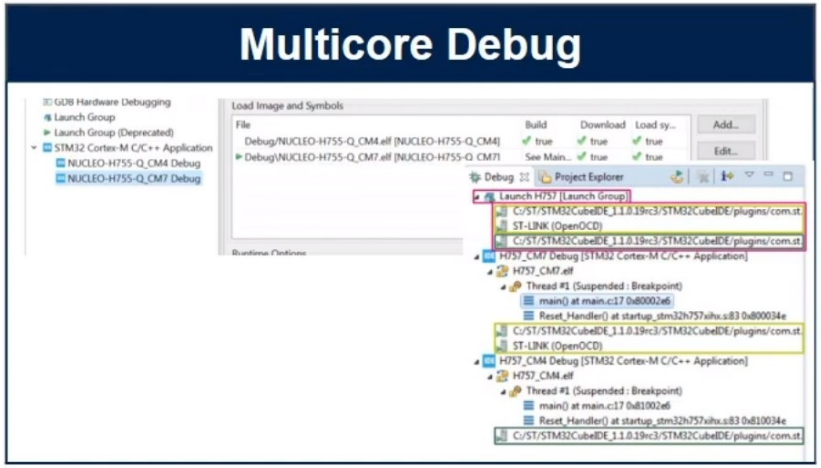
☑ Enable live expressio

☐ Log to file:

☑ External Loader: 512W3A_STM3210E-EVAL, 0x70000000, NAND_FLASH, 512W3A_STM3210E-EVAL.stldr

☐ Shared ST-LINK

```
512W3A_STM3210E-EVAL, 0x70000000, NAND_FLASH, 512W3A_STM3210E-EVAL.stldr
IS42S32800G_STM32769I-EVAL, 0xc0000000, SRAM, IS42S32800G_STM32769I-EVAL.stldr
IS61WV102416BLL_STM324x9I-EVAL, 0x64000000, SRAM, IS61WV102416BLL_STM324x9I-EVAL.stldr
IS61WV102416BLL_STM324xG-EVAL, 0x64000000, SRAM, IS61WV102416BLL_STM324xG-EVAL.stldr
IS61WV102416BLL_STM32769I-EVAL, 0x68000000, SRAM, IS61WV102416BLL_STM32769I-EVAL.stldr
IS61WV51216BLL_STM3210E-EVAL, 0x68000000, SRAM, IS61WV51216BLL_STM3210E-EVAL.stldr
IS66WV51216EBLL_STM32723E-DISCO, 0x60000000, SRAM, IS66WV51216EBLL_STM32723E-DISCO.stldr
IS66WV51216EBLL_STM32F413H-DISCO, 0x60000000, SRAM, IS66WV51216EBLL_STM32F413H-DISCO.stldr
M24LR-A_STM32303C-EVAL, 0x0, I2C_EEPROM, M24LR-A_STM32303C-EVAL.stldr
M24LR-A_STM32373C-EVAL, 0x0, I2C_EEPROM, M24LR-A_STM32373C-EVAL.stldr
M24LR-A_STM324x9I-EVAL, 0x0, I2C_EEPROM, M24LR-A_STM324x9I-EVAL.stldr
M25P64_STM3210E-EVAL, 0x0, SPI_FLASH, M25P64_STM3210E-EVAL.stldr
M29EWLA_STM32469I-EVAL, 0x60000000, NOR_FLASH, M29EWLA_STM32469I-EVAL.stldr
M29W128GL_STM3210E-EVAL, 0x64000000, NOR_FLASH, M29W128GL_STM3210E-EVAL.stldr
M29W128GL_STM324x9I-EVAL, 0x60000000, NOR_FLASH, M29W128GL_STM324x9I-EVAL.stldr
M29W128GL_STM32F756G-EVAL, 0x60000000, NOR_FLASH, M29W128GL_STM32F756G-EVAL.stldr
M29W128GL_STM32F756G-EVAL2, 0x60000000, NOR_FLASH, M29W128GL_STM32F756G-EVAL2.stldr
```



Multicore Debug

- GDB Hardware Debugging
- Launch Group
- Launch Group (Deprecated)
- STM32 Cortex-M C/C++ Application
  - NUCLEO-H755-Q_CM4 Debug
  - NUCLEO-H755-Q_CM7 Debug

Load Image and Symbols

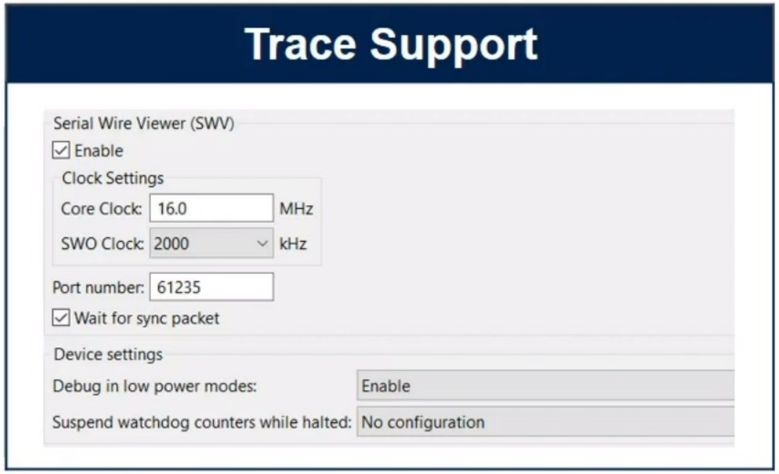
File	Build	Download	Load sy...
Debug/NUCLEO-H755-Q_CM4.elf [NUCLEO-H755-Q_CM4]	✓ true	✓ true	✓ true
▶ Debug\NUCLEO-H755-Q_CM7.elf [NUCLEO-H755-Q_CM7]	See Main...	✓ true	✓ true

Add... | Edit...

Runtime Options

Debug | Project Explorer

```
Launch H757 [Launch Group]
    C:/ST/STM32CubeIDE_1.1.0.19rc3/STM32CubeIDE/plugins/com.st.
    ST-LINK (OpenOCD)
    C:/ST/STM32CubeIDE_1.1.0.19rc3/STM32CubeIDE/plugins/com.st.
H757_CM7 Debug [STM32 Cortex-M C/C++ Application]
  H757_CM7.elf
    Thread #1 (Suspended : Breakpoint)
      main() at main.c:17 0x80002e6
      Reset_Handler() at startup_stm32h757xihx.s:83 0x800034e
    C:/ST/STM32CubeIDE_1.1.0.19rc3/STM32CubeIDE/plugins/com.st.
    ST-LINK (OpenOCD)
H757_CM4 Debug [STM32 Cortex-M C/C++ Application]
  H757_CM4.elf
    Thread #1 (Suspended : Breakpoint)
      main() at main.c:17 0x81002e6
      Reset_Handler() at startup_stm32h757xihx.s:83 0x810034e
    C:/ST/STM32CubeIDE_1.1.0.19rc3/STM32CubeIDE/plugins/com.st.
```



Trace Support

Serial Wire Viewer (SWV)

☑ Enable

Clock Settings

Core Clock: 16.0 MHz

SWO Clock: 2000 kHz

Port number: 61235

☑ Wait for sync packet

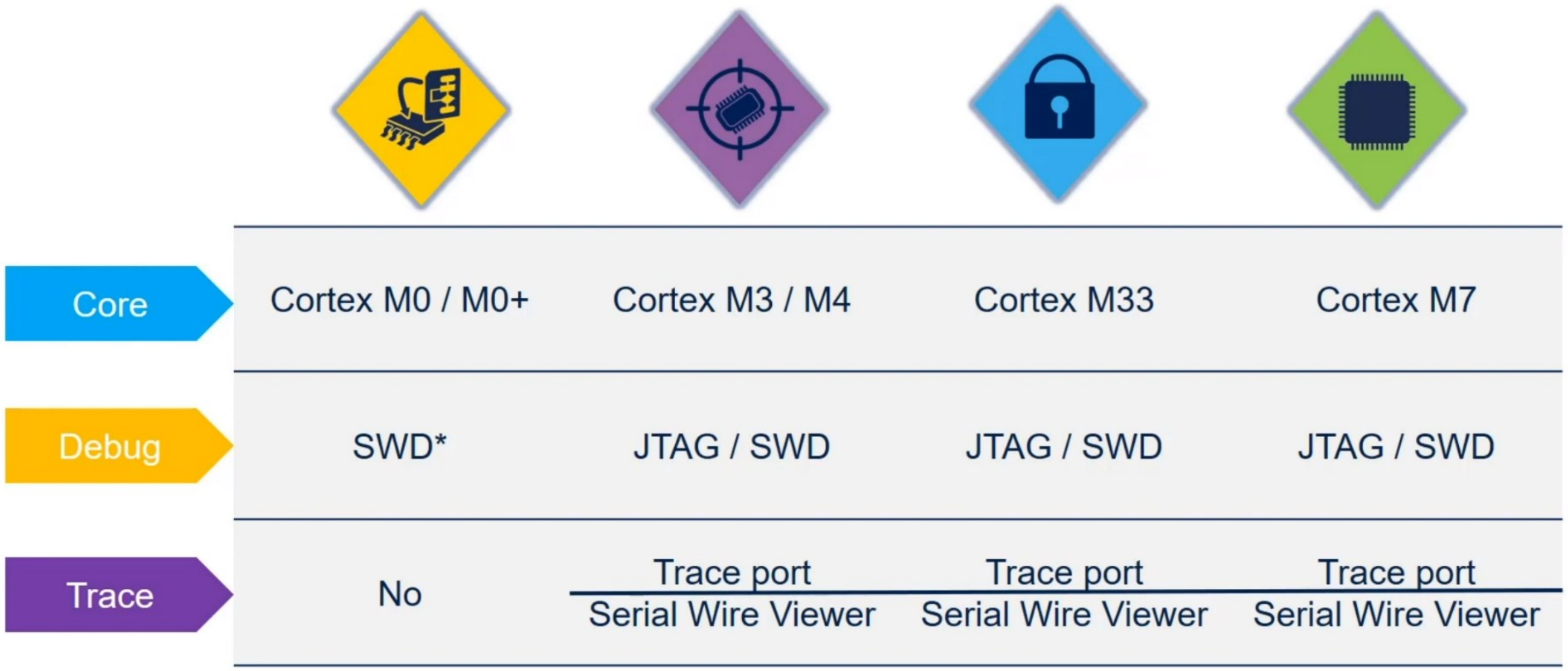
Device settings

Debug in low power modes: Enable

Suspend watchdog counters while halted: No configuration

2. Programiranje vgrajenih sistemov

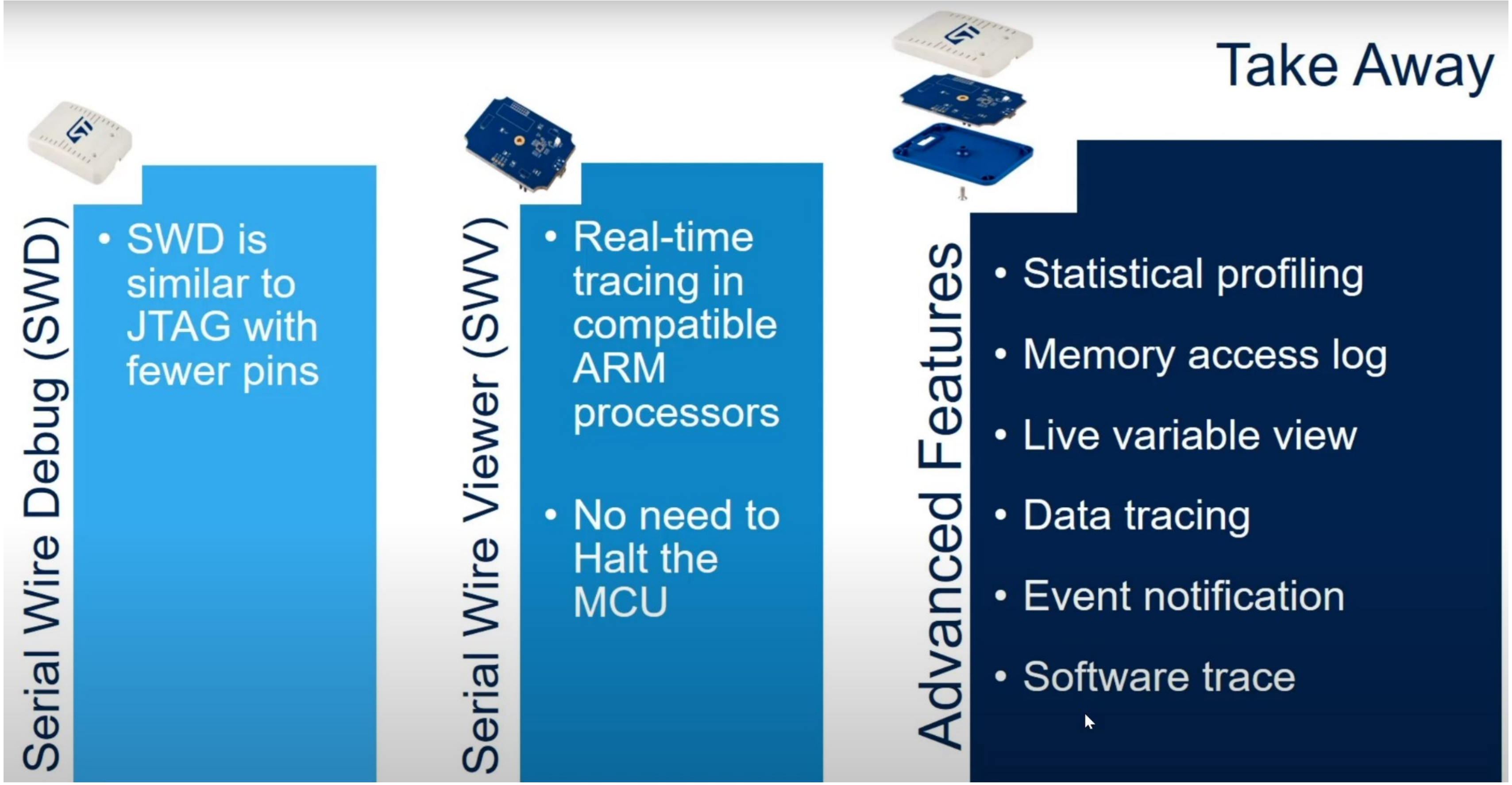
Razvoj in razhroščevanje (primer CubeIDE)



Core	Cortex M0 / M0+	Cortex M3 / M4	Cortex M33	Cortex M7
Debug	SWD*	JTAG / SWD	JTAG / SWD	JTAG / SWD
Trace	No	Trace port Serial Wire Viewer	Trace port Serial Wire Viewer	Trace port Serial Wire Viewer

2. Programiranje vgrajenih sistemov

Razvoj in razhroščevanje (primer CubeIDE)



Take Away

Serial Wire Debug (SWD)

- SWD is similar to JTAG with fewer pins

Serial Wire Viewer (SWV)

- Real-time tracing in compatible ARM processors
- No need to Halt the MCU

Advanced Features

- Statistical profiling
- Memory access log
- Live variable view
- Data tracing
- Event notification
- Software trace

2. Programiranje vgrajenih sistemov

Razvoj in razhroščevanje (primer CubeIDE)

2. Programiranje vgrajenih sistemov

Razvoj in razhroščevanje (primer CubeIDE)

Razvoj in razhroščevanje (primer CubeIDE)

- Using Data Trace & Live Watch
- Timing Measurement
- Using Data Trace Timeline Graph
- Exception Log & Timeline Graph
- printf() redirection
- Statistical Profiling

- SWV adds:

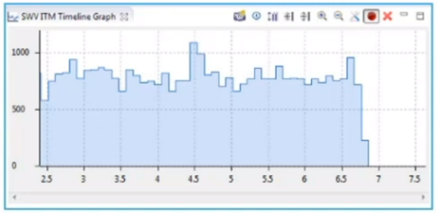
Real Time Trace

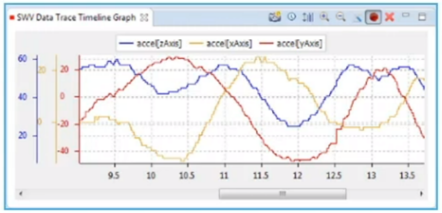
- That uses the SWD port and the SWO pin.

Advanced Debugging

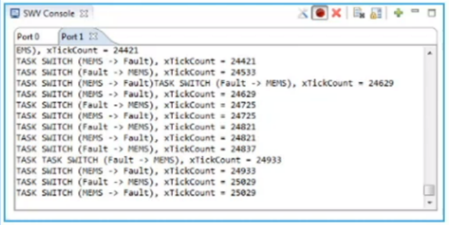
- Without Halting the MCU

Plots

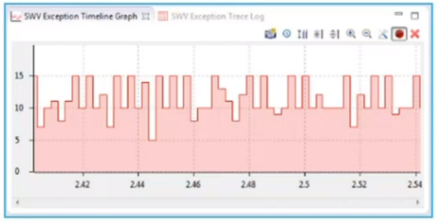




Logs



```
EMS), xTickCount = 24421
TASK SWITCH (MEMS -> Fault), xTickCount = 24421
TASK SWITCH (Fault -> MEMS), xTickCount = 24533
TASK SWITCH (MEMS -> Fault)TASK SWITCH (Fault -> MEMS), xTickCount = 24629
TASK SWITCH (MEMS -> Fault), xTickCount = 24629
TASK SWITCH (Fault -> MEMS), xTickCount = 24725
TASK SWITCH (MEMS -> Fault), xTickCount = 24725
TASK SWITCH (Fault -> MEMS), xTickCount = 24821
TASK SWITCH (MEMS -> Fault), xTickCount = 24821
TASK SWITCH (Fault -> MEMS), xTickCount = 24837
TASK TASK SWITCH (Fault -> MEMS), xTickCount = 24933
TASK SWITCH (MEMS -> Fault), xTickCount = 24933
TASK SWITCH (Fault -> MEMS), xTickCount = 25029
TASK SWITCH (MEMS -> Fault), xTickCount = 25029
```



2. Programiranje vgrajenih sistemov

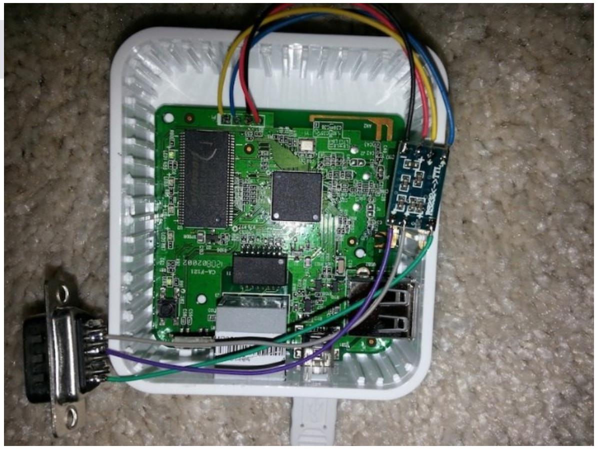
Razvoj in razhroščevanje (primer CubeIDE)

2. Programiranje vgrajenih sistemov

Razvoj in razhroščevanje (primer CubeIDE)

2. Programiranje vgrajenih sistemov

Kaj po koncu razvoja ?



- Dokumentacija (!*?)

- Spremljanje delovanja (kurativa) :

spremljanje

  - serijska konzola
  - log datoteke, sporočanje napak, daljinski nadzor
    - primer kritične napake:

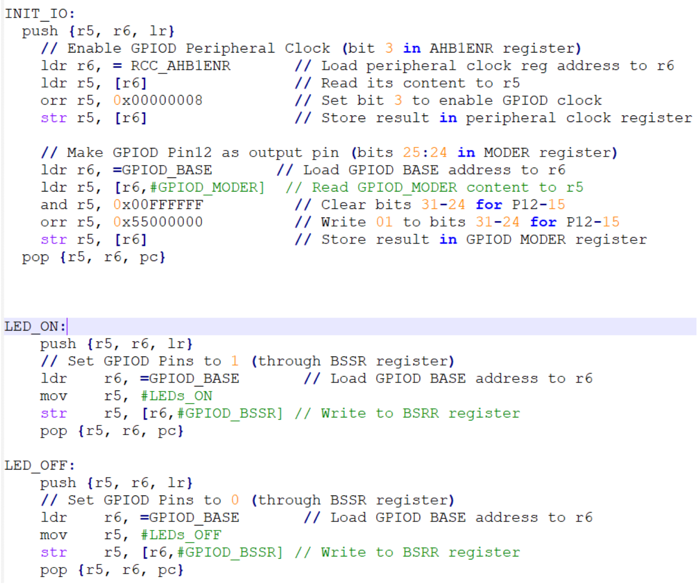
```
2015-01-11 05:45:37 CRIT   232        0 FP       WDT has expired
2015-01-11 05:46:21 CRIT   232        0 MNG      WDT has expired
```

    - primer pomembne napake, ki zahteva popravke v kodi :

```
2015-01-09 15:00:02 INFO    60        0 CMDEXECUTE  CMD:Execute Cmd[72]
2015-01-09 15:00:02 INFO    60        0 CMDEXECUTE CMD:SendSett
2015-01-09 15:04:02 CRIT   232        0 CMDEXECUTE  WDT has expired
```

3. Nivoji programiranja – jeziki, knjižnice

Baremetal - zbirnik



```
INIT_IO:
  push {r5, r6, lr}
    // Enable GPIOD Peripheral Clock (bit 3 in AHB1ENR register)
    ldr r6, = RCC_AHB1ENR       // Load peripheral clock reg address to r6
    ldr r5, [r6]                // Read its content to r5
    orr r5, 0x00000008          // Set bit 3 to enable GPIOD clock
    str r5, [r6]                // Store result in peripheral clock register

    // Make GPIOD Pin12 as output pin (bits 25:24 in MODER register)
    ldr r6, =GPIOD_BASE       // Load GPIOD BASE address to r6
    ldr r5, [r6,#GPIOD_MODER]  // Read GPIOD_MODER content to r5
    and r5, 0x00FFFFFF          // Clear bits 31-24 for P12-15
    orr r5, 0x55000000          // Write 01 to bits 31-24 for P12-15
    str r5, [r6]                // Store result in GPIOD MODER register
  pop {r5, r6, pc}



LED_ON:
    push {r5, r6, lr}
    // Set GPIOD Pins to 1 (through BSSR register)
    ldr    r6, =GPIOD_BASE        // Load GPIOD BASE address to r6
    mov    r5, #LEDs_ON
    str    r5, [r6,#GPIOD_BSSR] // Write to BSRR register
    pop {r5, r6, pc}

LED_OFF:
    push {r5, r6, lr}
    // Set GPIOD Pins to 0 (through BSSR register)
    ldr    r6, =GPIOD_BASE        // Load GPIOD BASE address to r6
    mov    r5, #LEDs_OFF
    str    r5, [r6,#GPIOD_BSSR] // Write to BSRR register
    pop {r5, r6, pc}
```

https://github.com/LAPSyLAB/ORLab-STM32/tree/main/GPIO_LEDs

Baremetal - C

```
   /* USER CODE BEGIN 2 */


  RCC->AHB1ENR |= 0x08;
// Enable clock for GPIOD
  GPIOD->MODER |= 0x01000000;          //
MODE Register: bit 12 == out

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    GPIOD->ODR ^= 0x1000;              //
Toggle PD12

/* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    for (int i=0; i<0x1000000; i++) {};
// waste some time
  }
  /* USER CODE END 3 */
```

https://github.com/LAPSyLAB/STM32F4_Discovery_VIN_Projects/tree/main/LED_GPIO_C_Baremetal_C

HAL - C

```
/* Infinite loop */
 /* USER CODE BEGIN WHILE */
 while (1)
 {

HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_12);

   /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
 HAL_Delay(1000);
 }
 /* USER CODE END 3 */



void HAL_GPIO_TogglePin(GPIO_TypeDef* GPIOx,
uint16_t GPIO_Pin)
{
  uint32_t odr;

  /* Check the parameters */
  assert_param(IS_GPIO_PIN(GPIO_Pin));

  /* get current Ouput Data Register value
*/
  odr = GPIOx->ODR;

  /* Set selected pins that were at low
level, and reset ones that were high */
  GPIOx->BSRR = ((odr & GPIO_Pin) <<
GPIO_NUMBER) | (~odr & GPIO_Pin);
}
```

https://github.com/LAPSyLAB/STM32F4_Discovery_VIN_Projects/tree/main/LED_Blink_Demo

3. Nivoji programiranja – koncepti

Ena zanka

```
{ …
        if (Timer_1sec)   {
                readSensors(&data);
                send_data(&data);
                 Timer_1sec = 0;
        }

        if (Timer_50msec)   {
                readKeys(&keys);
                readInputs(&inputs);
                Timer_50msec = 0;
        }
}
```

Končni avtomat

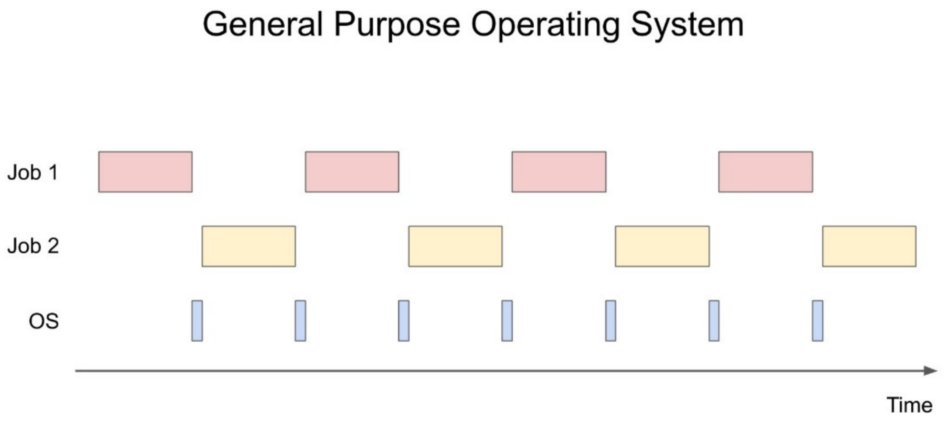
```
switch (FSM.State) {
        case CHECK_REASON:
            ///<  FSM.State: after reset.

            if  VSE_OK then
               FSM.State  = CHECK_BAUDRATE

               break;

        case CHECK_BAUDRATE:
                 ///<  FSM.State: after reset.
                   …

               break;
```

RTOS

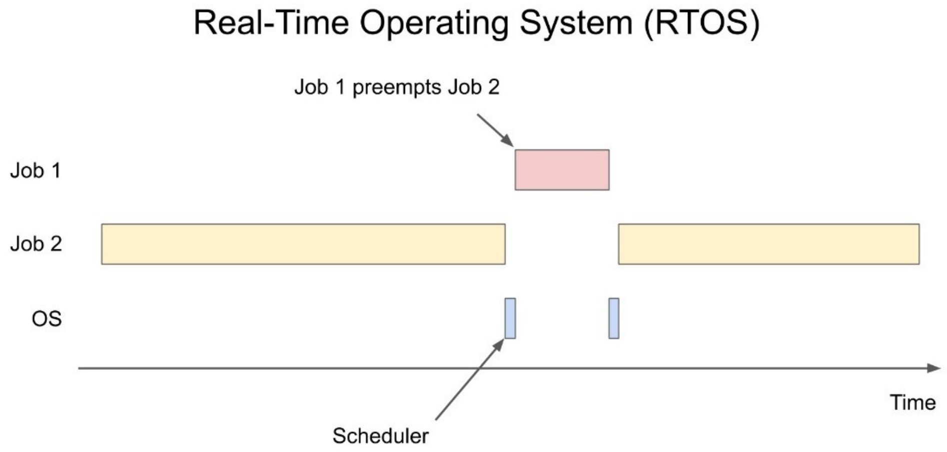
```
void StartTask02(void *argument)
{
  /* USER CODE BEGIN StartTask02 */
  /* Infinite loop */
  for(;;)
  {
   HAL_GPIO_TogglePin(GPIOD,
GPIO_PIN_13);
    osDelay(1000);
  }
  /* USER CODE END StartTask02 */
}

void StartTask01(void *argument)
{
  /* USER CODE BEGIN StartTask01 */
  /* Infinite loop */
  for(;;)
  {
HAL_GPIO_TogglePin(GPIOD,
GPIO_PIN_12);

    osDelay(1000);

  }
  /* USER CODE END StartTask01 */
}
```

4. Programiranje vgrajenih sistemov – OS, RTOS

OS



RTOS



4.1 Splošno o RTOS - Real Time Operating System

- RTOS upravlja čas in procese na mikroprocesorju ali mikrokrmilniku
  - v bistvu: „poenostavljen operacijski sistem“

- Funkcionalnosti RTOS:
  - Večopravilnost (multi-tasking)
  - Dodeljevanje opravil CPE s prioritetami
  - Sinhronizacija dostopov do virov:
    - V/I naprav
    - Pomnilnika (podatkovnih struktur)
  - Komunikacija med procesi (Inter-task communication)
  - Časovna predvidljivost (realno-časna odzivnost)
  - Servisiranje prekinitev

Zakaj uporabiti RTOS?

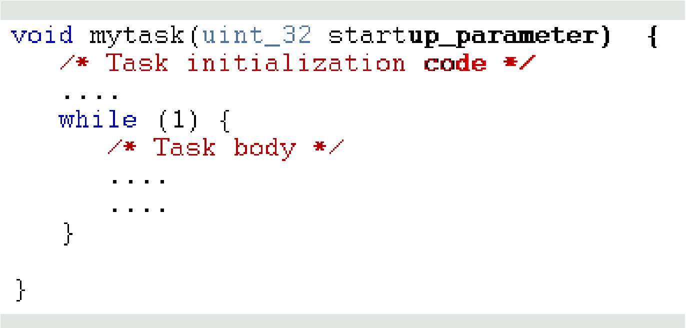
- Uporaba V/I naprav (že pripravljeni driverji (TCP, ETH, CANBUS,…)
- Se splača vse razviti iz nič (npr. svoj dodeljevalnik) ?
  - Diploma : Fabčič – 2021 – lasten RTOS „from scratch“
- Večopravilnost z možnostjo sinhronizacije
- Prenosljivost kode na druge CPE
- Upravljanje z viri
- Možnost dopolnitve z lastnimi funkcijami
- Obstoječa podpora za nekatere razširjene protokole:
  - TCP/IP, USB, Flash Systems, Web Servers,
  - CAN protocols, GUI, SSL, SNMP

Nekatere prednosti se hitro sprevržejo v težave in dodatno delo…

RTOS - Opravila

- Sistem oz. aplikacija je sestavljena iz več opravil
- Opravila se izmenjaje izvajajo
- V nekem trenutku je aktivno natanko eno opravilo (se izvaja na procesorju)
- RTOS odloča, kako si opravila delijo procesor („context switching“)
- Vsebina opravila („Task Context“)
  - Podatkovna struktura lastna vsakemu opravilu:
    - Vsebuje vse potrebne podatke za izvedbo opravila:
      - □ npr. spremenljivke, registre in sezname vseh uporabljenih virov

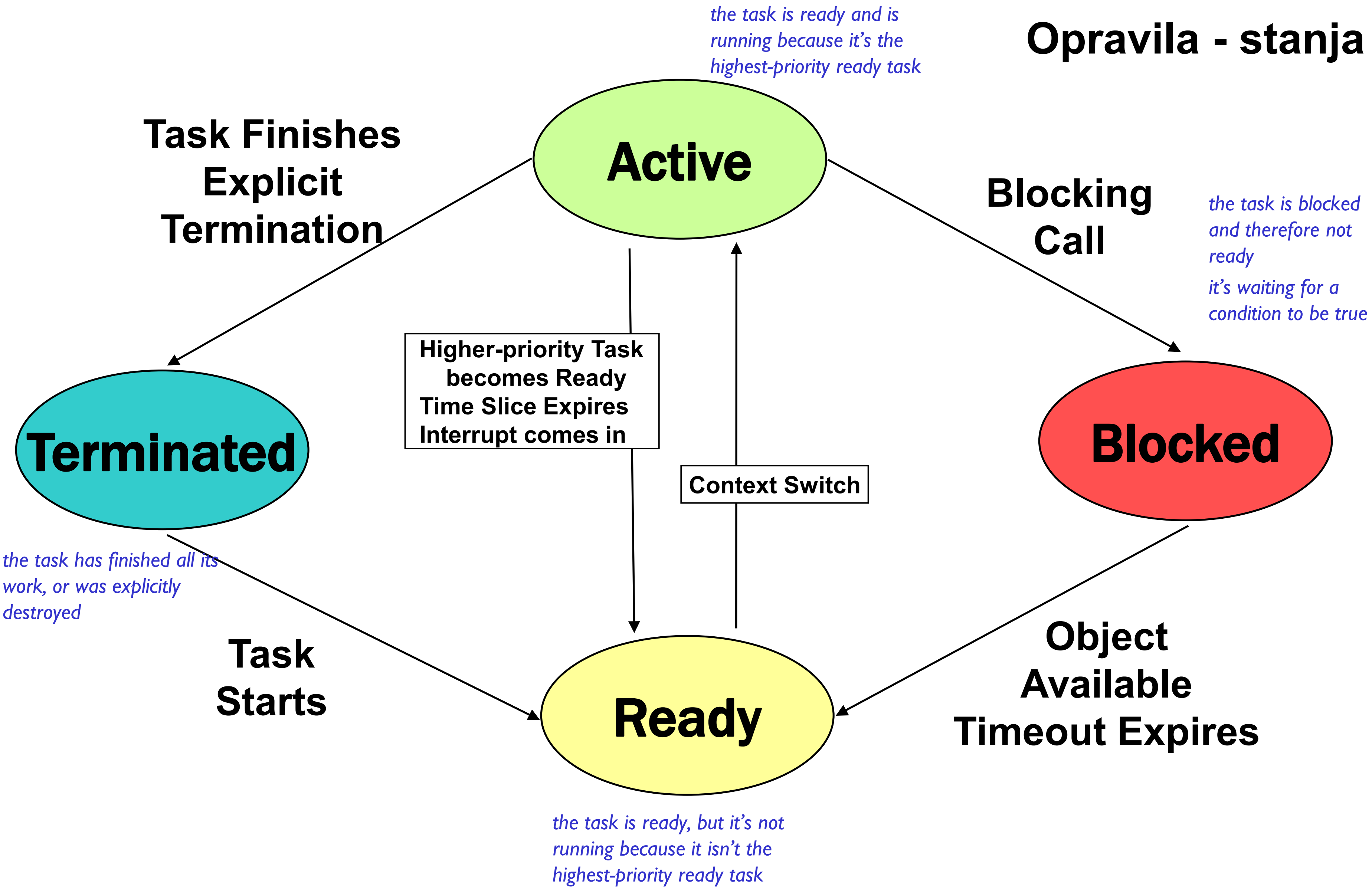
Tipična struktura kode opravila



```
void mytask(uint_32 startup_parameter)  {
   /* Task initialization code */
   ....
   while (1) {
      /* Task body */
      ....
      ....
   }

}
```

```
void StartTask02(void *argument)
{
  /* USER CODE BEGIN StartTask02 */
  /* Infinite loop */
  for(;;)
  {
   HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_13);
    osDelay(1000);
  }
  /* USER CODE END StartTask02 */
}

void StartTask01(void *argument)
{
  /* USER CODE BEGIN StartTask01 */
  /* Infinite loop */
  for(;;)
  {
HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_12);

    osDelay(1000);

  }
  /* USER CODE END StartTask01 */
}
```

Opravila - stanja



the task is ready and is running because it's the highest-priority ready task

Task Finishes
Explicit
Termination

Blocking
Call

the task is blocked and therefore not ready

it's waiting for a condition to be true

Higher-priority Task becomes Ready
Time Slice Expires
Interrupt comes in

Context Switch

the task has finished all its work, or was explicitly destroyed

Task
Starts

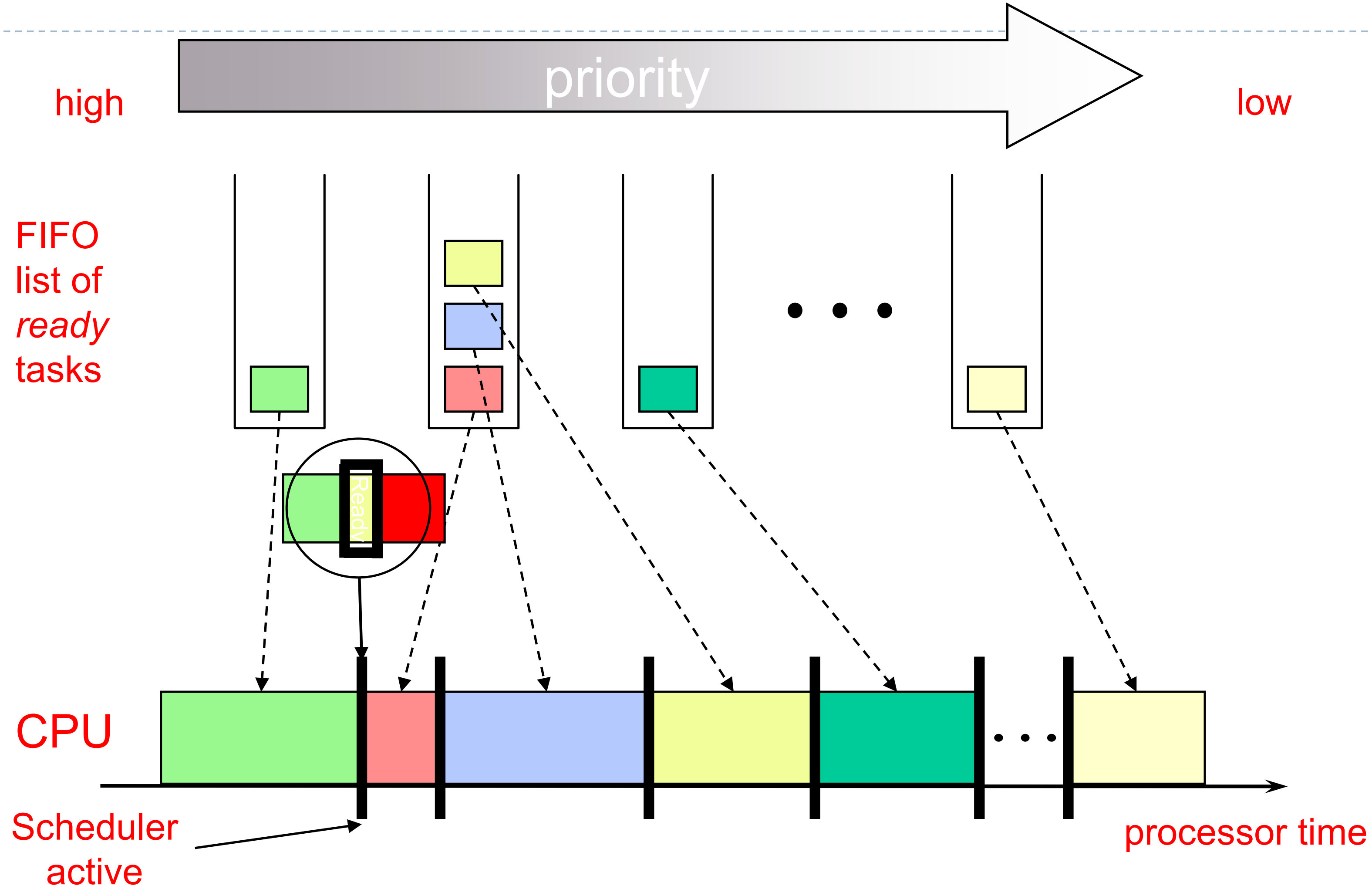
Object
Available
Timeout Expires

the task is ready, but it's not running because it isn't the highest-priority ready task

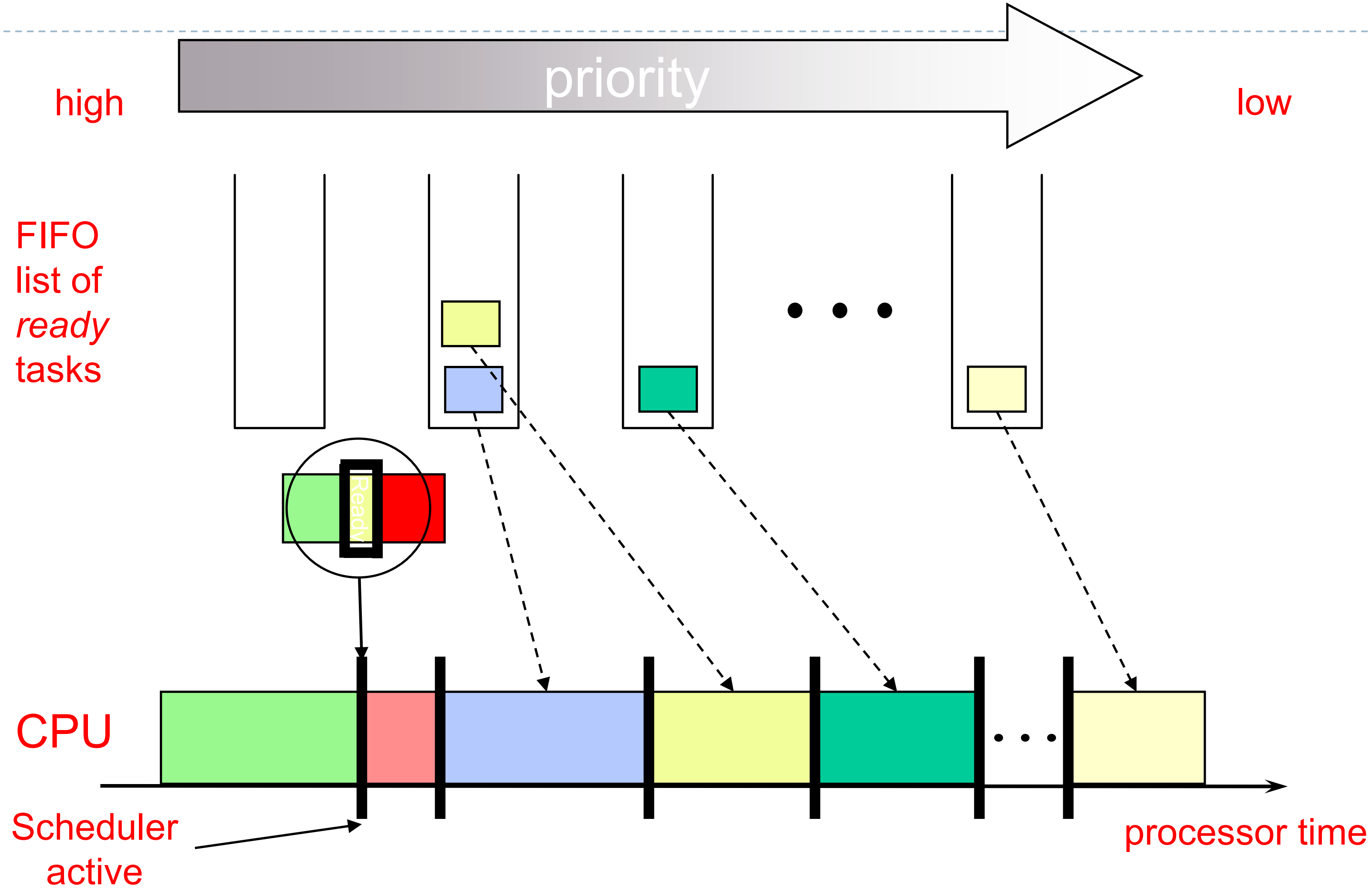
Dodeljevalnik („Scheduler“)

- **Običajni načini dodeljevanja:**
  - FIFO (tudi „priority-based preemptive“)
    - Aktivni je tisti z najvišjo prioriteto, ki je pripravljen najdlje časa
  - Round Robin
    - Aktivni je tisti z najvišjo prioriteto, ki je najdlje časa brez dodelitve procesorju

Priority Based FIFO Scheduling



Priority Based FIFO Scheduling



high

priority

low

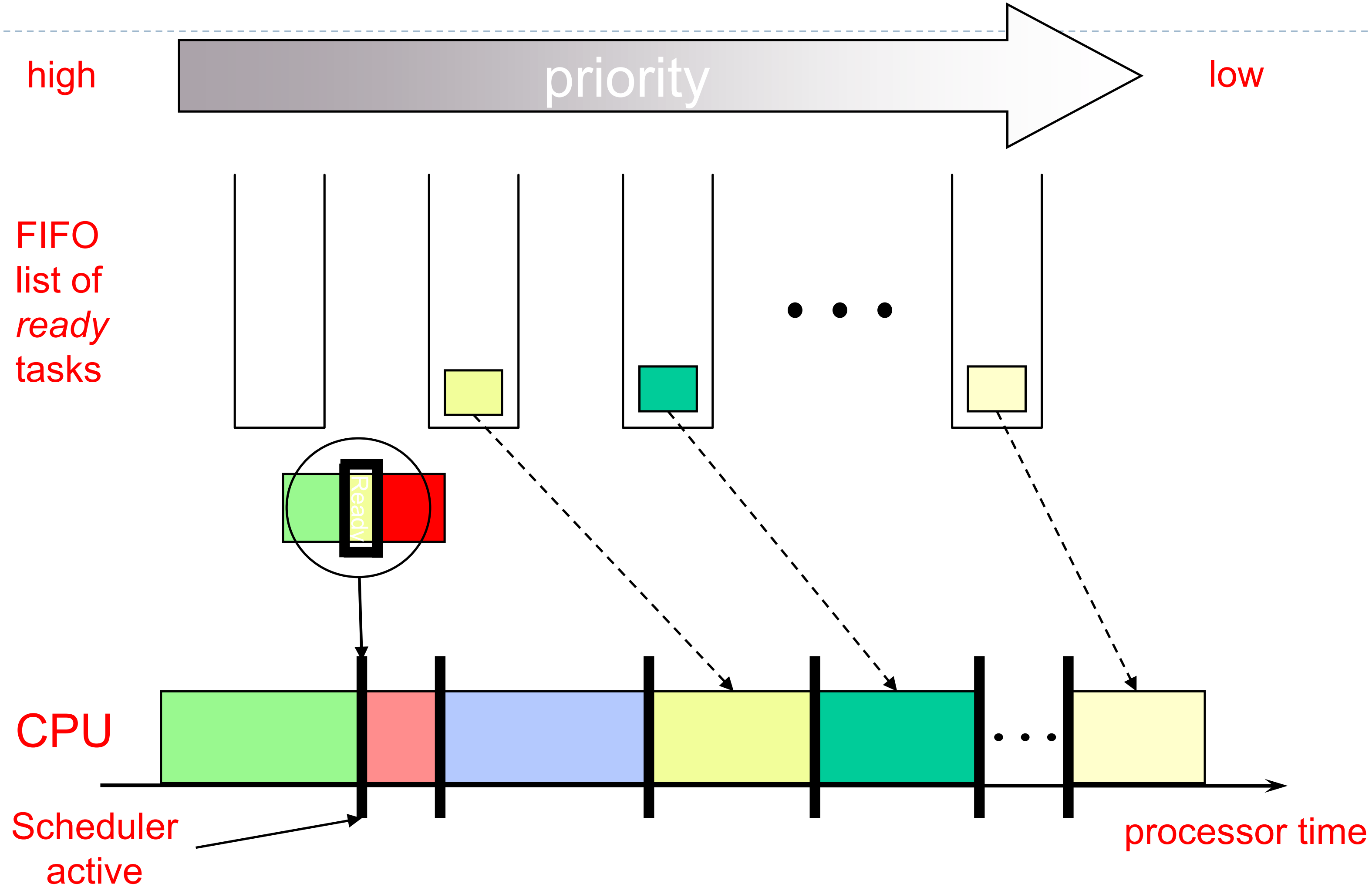
FIFO list of *ready* tasks

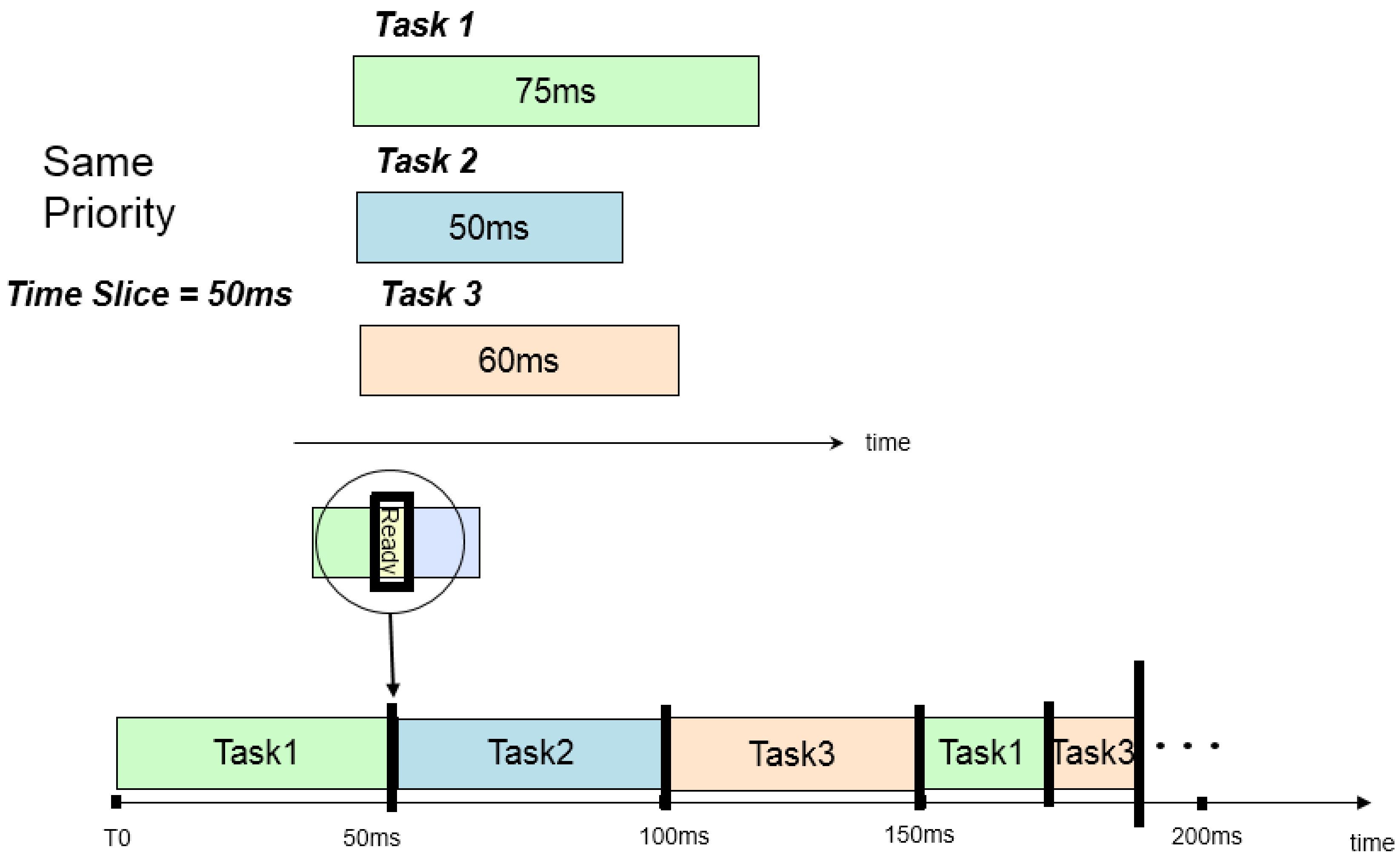
CPU

Scheduler active

processor time

Priority Based FIFO Scheduling





Same Priority

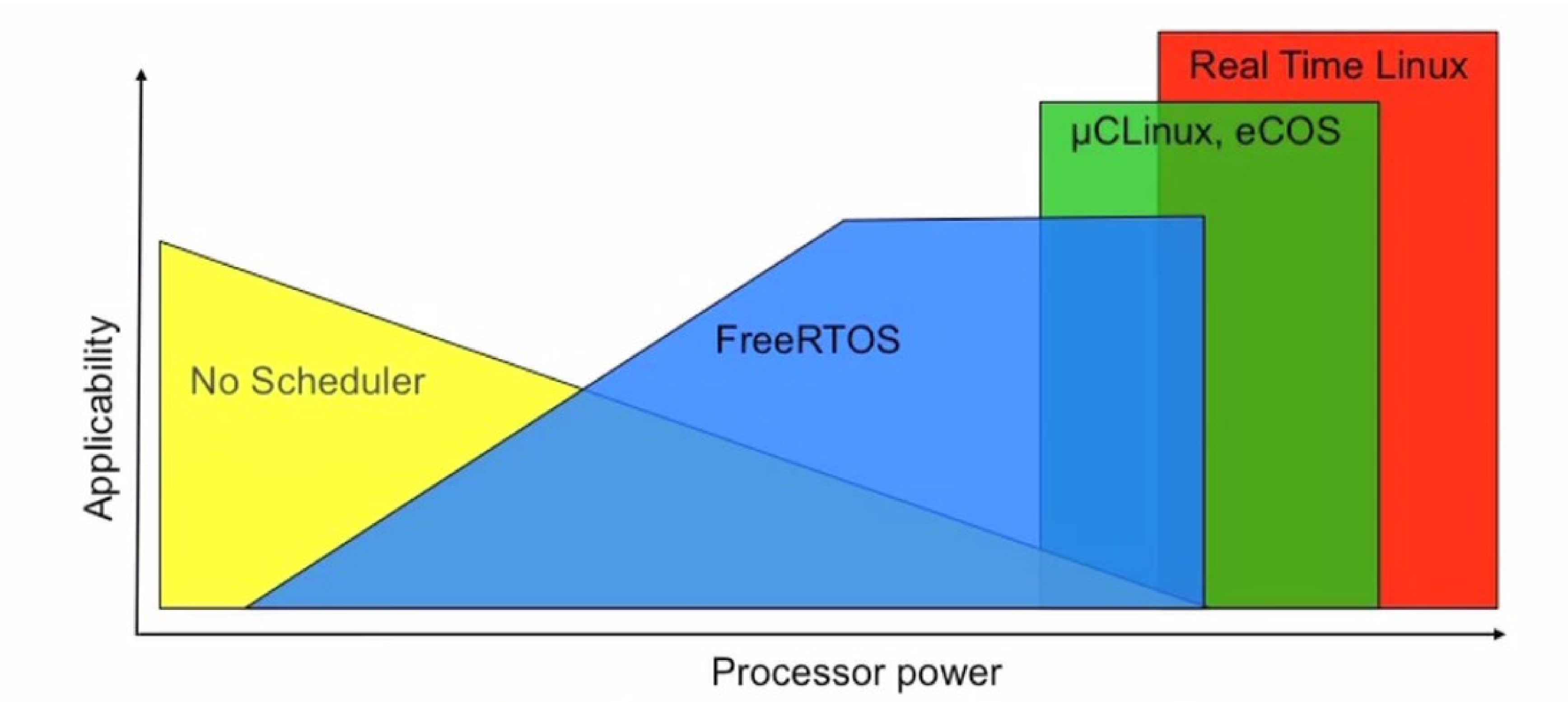
Time Slice = 50ms

4.2. FreeRTOS (primer) :

- A Real Time Operating System
- Written by Richard Barry & FreeRTOS Team

- Huge number of users all over the world
  - 6000 Download per month
- Simple but very powerful

Kdaj uporabiti FreeRTOS ?



4.2 FreeRTOS (primer) :

Opravila („Tasks“) - Primer

```
/**
* @brief Function implementing the ShellTask thread.
* @param argument: Not used
* @retval None
*/
/* USER CODE END Header_Shell_Entry */
void Shell_Entry(void const * argument)
{
  /* USER CODE BEGIN Shell_Entry */
            printf_dma ("\r\nShell Task started.\r\n");
            if ( HAL_UART_Receive_IT(&huart3, &(UARTRxBuffer[Var.Uart.RxBufferInd]), 1) != HAL_OK) {
                    Error_Handler();
             }

            shell_cmd_init(); ///< Init command shell

            /* Infinite loop */
            for(;;)
            {
                  shell_cmd_check_rx();     ///< check if shell character received
                  osDelay(100);
                  ShelluxHighWaterMark = uxTaskGetStackHighWaterMark( NULL );
            }
  /* USER CODE END Shell_Entry */
}
```

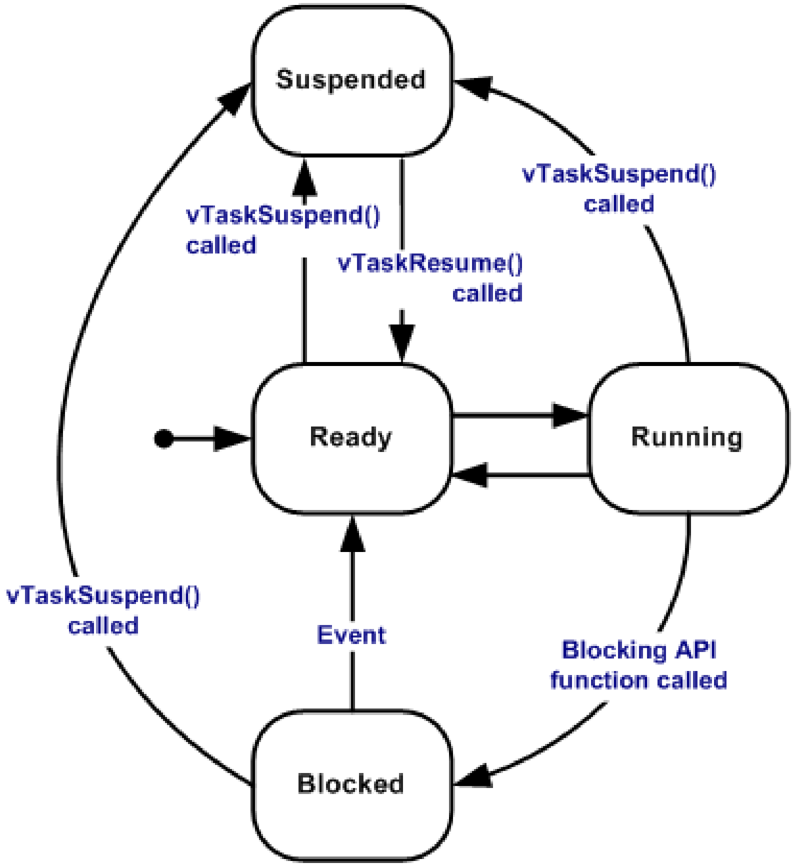
4.2. FreeRTOS (primer) :

Architecture Overview

- Tasks
  - task.c , task.h
    - creating, scheduling, and maintaining tasks.
- Communication
  - queue.c and queue.h handle communication. Tasks and interrupts use queues to
    - send data to each other and
    - to signal the use of critical resources using semaphores and mutexes.
- Hardware Interfacing

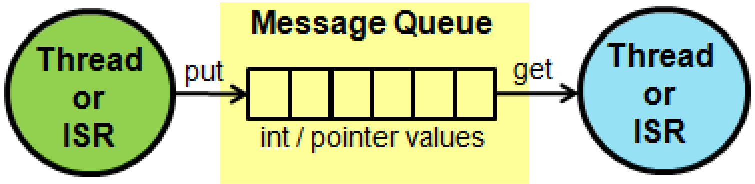
Stanje procesov

- Running
- Ready
- Blocked
- Suspended

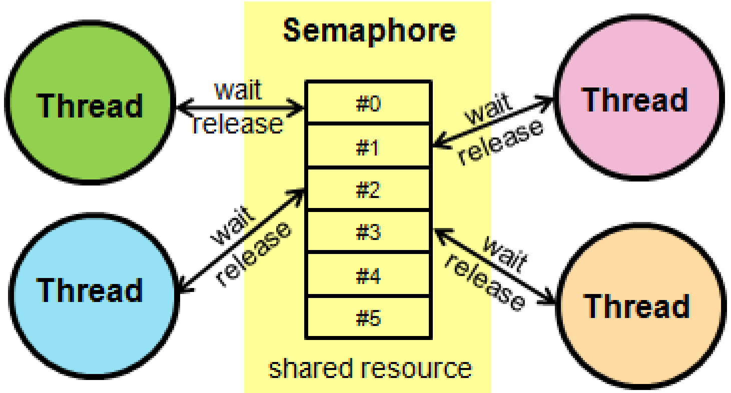


__RTOS : Komunikacija in sinhronizacija med procesi

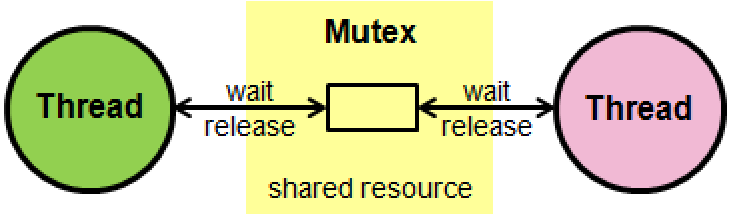
- Queues



- Binary Semaphores
- Counting Semaphores



- Mutexes
- Recursive Mutexes

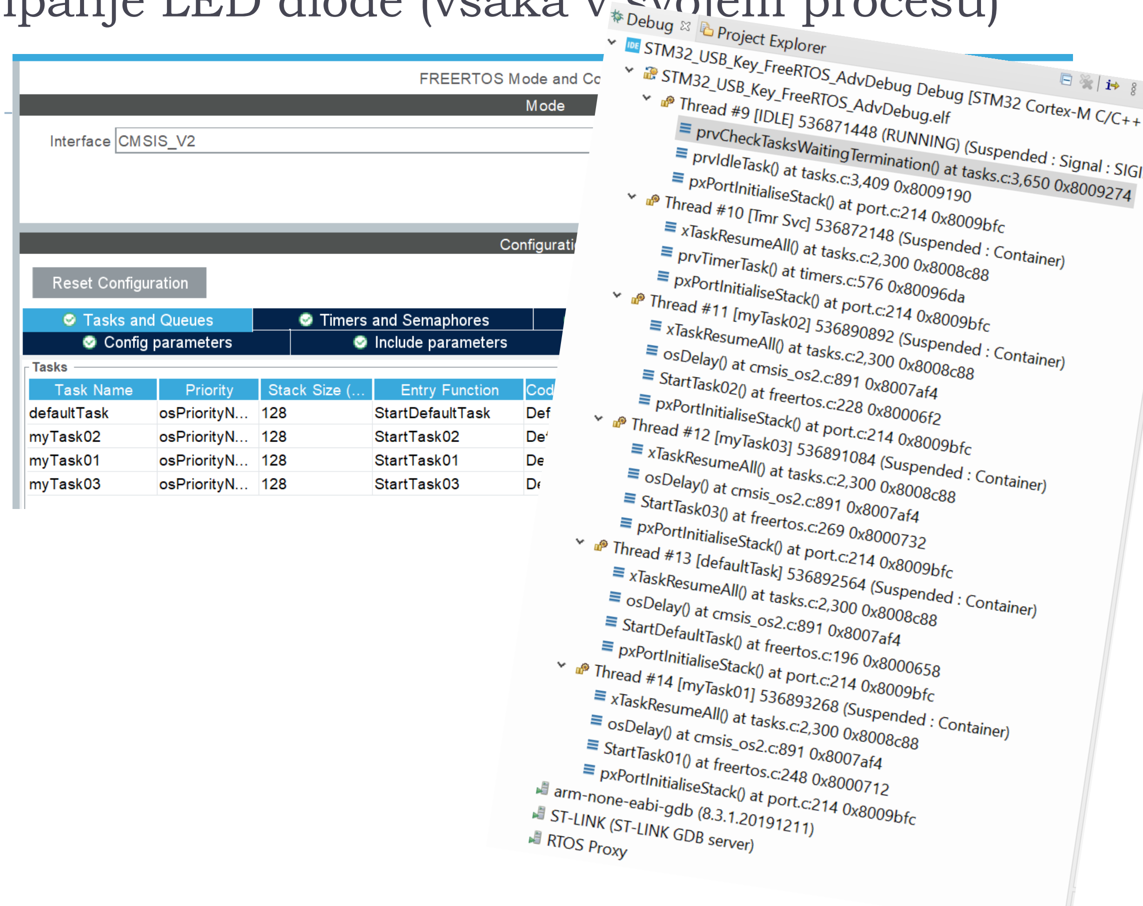


FreeRTOS uradna spletna stran:
http://www.freertos.org/

4.2. FreeRTOS (primer) : Utripanje LED diode (vsaka v svojem procesu)

Opravila („Tasks“)

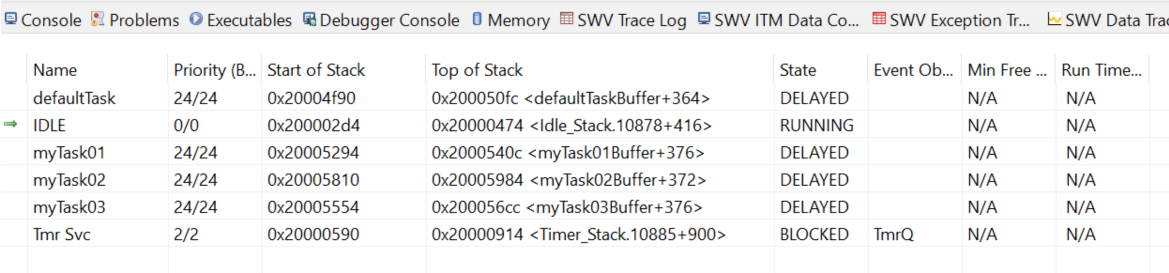
```
void StartTask02(void *argument)
{
  /* USER CODE BEGIN StartTask02 */
  /* Infinite loop */
  for(;;)
  {
   HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_13);
    osDelay(1000);
  }
  /* USER CODE END StartTask02 */
}

void StartTask01(void *argument)
{
  /* USER CODE BEGIN StartTask01 */
  /* Infinite loop */
  for(;;)
  {
    HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_12);

    osDelay(1000);

  }
  /* USER CODE END StartTask01 */
}
```



Interface CMSIS_V2

Task Name	Priority	Stack Size (...	Entry Function	Cod
defaultTask	osPriorityN...	128	StartDefaultTask	Def
myTask02	osPriorityN...	128	StartTask02	De
myTask01	osPriorityN...	128	StartTask01	De
myTask03	osPriorityN...	128	StartTask03	De

```
STM32_USB_Key_FreeRTOS_AdvDebug Debug [STM32 Cortex-M C/C++
  STM32_USB_Key_FreeRTOS_AdvDebug.elf
    Thread #9 [IDLE] 536871448 (RUNNING) (Suspended : Signal : SIGI
      prvCheckTasksWaitingTermination() at tasks.c:3,650 0x8009274
      prvIdleTask() at tasks.c:3,409 0x8009190
      pxPortInitialiseStack() at port.c:214 0x8009bfc
    Thread #10 [Tmr Svc] 536872148 (Suspended : Container)
      xTaskResumeAll() at tasks.c:2,300 0x8008c88
      prvTimerTask() at timers.c:576 0x80096da
      pxPortInitialiseStack() at port.c:214 0x8009bfc
    Thread #11 [myTask02] 536890892 (Suspended : Container)
      xTaskResumeAll() at tasks.c:2,300 0x8008c88
      osDelay() at cmsis_os2.c:891 0x8007af4
      StartTask02() at freertos.c:228 0x80006f2
      pxPortInitialiseStack() at port.c:214 0x8009bfc
    Thread #12 [myTask03] 536891084 (Suspended : Container)
      xTaskResumeAll() at tasks.c:2,300 0x8008c88
      osDelay() at cmsis_os2.c:891 0x8007af4
      StartTask03() at freertos.c:269 0x8000732
      pxPortInitialiseStack() at port.c:214 0x8009bfc
    Thread #13 [defaultTask] 536892564 (Suspended : Container)
      xTaskResumeAll() at tasks.c:2,300 0x8008c88
      osDelay() at cmsis_os2.c:891 0x8007af4
      StartDefaultTask() at freertos.c:196 0x8000658
      pxPortInitialiseStack() at port.c:214 0x8009bfc
    Thread #14 [myTask01] 536893268 (Suspended : Container)
      xTaskResumeAll() at tasks.c:2,300 0x8008c88
      osDelay() at cmsis_os2.c:891 0x8007af4
      StartTask01() at freertos.c:248 0x8000712
      pxPortInitialiseStack() at port.c:214 0x8009bfc
  arm-none-eabi-gdb (8.3.1.20191211)
  ST-LINK (ST-LINK GDB server)
  RTOS Proxy
```



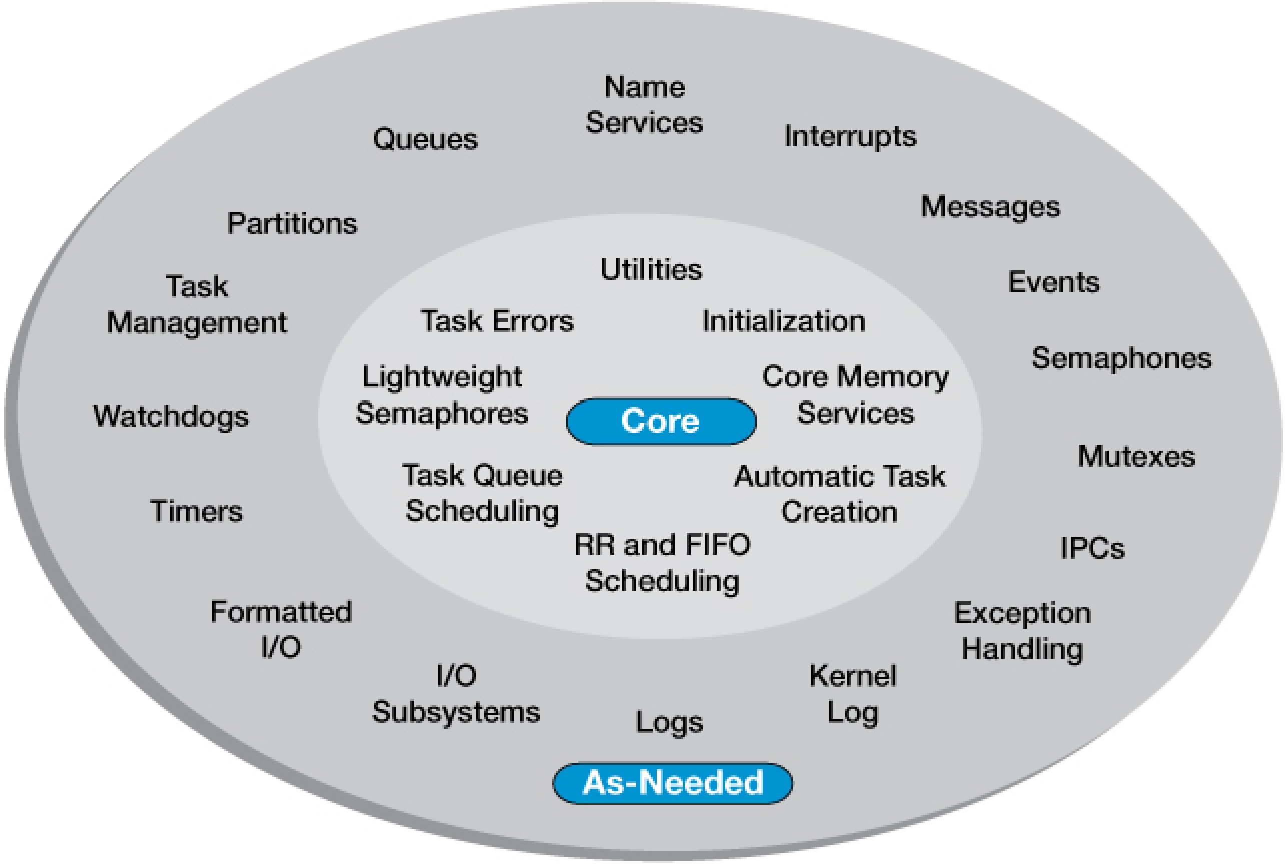
Name	Priority (B...	Start of Stack	Top of Stack	State	Event Ob...	Min Free ...	Run Time...
defaultTask	24/24	0x20004f90	0x200050fc <defaultTaskBuffer+364>	DELAYED		N/A	N/A
⇒ IDLE	0/0	0x200002d4	0x20000474 <Idle_Stack.10878+416>	RUNNING		N/A	N/A
myTask01	24/24	0x20005294	0x2000540c <myTask01Buffer+376>	DELAYED		N/A	N/A
myTask02	24/24	0x20005810	0x20005984 <myTask02Buffer+372>	DELAYED		N/A	N/A
myTask03	24/24	0x20005554	0x200056cc <myTask03Buffer+376>	DELAYED		N/A	N/A
Tmr Svc	2/2	0x20000590	0x20000914 <Timer_Stack.10885+900>	BLOCKED	TmrQ	N/A	N/A

4.2. FreeRTOS (STM32F4 primer) :

Opravila („Tasks“)

4.3 MQX RTOS (primer) :

MQX™ RTOS: Customizable Component Set



Name Services

Queues

Interrupts

Messages

Partitions

Utilities

Events

Task Management

Task Errors

Initialization

Semaphones

Lightweight Semaphores

Core Memory Services

Watchdogs

Core

Mutexes

Task Queue Scheduling

Automatic Task Creation

Timers

RR and FIFO Scheduling

IPCs

Formatted I/O

Exception Handling

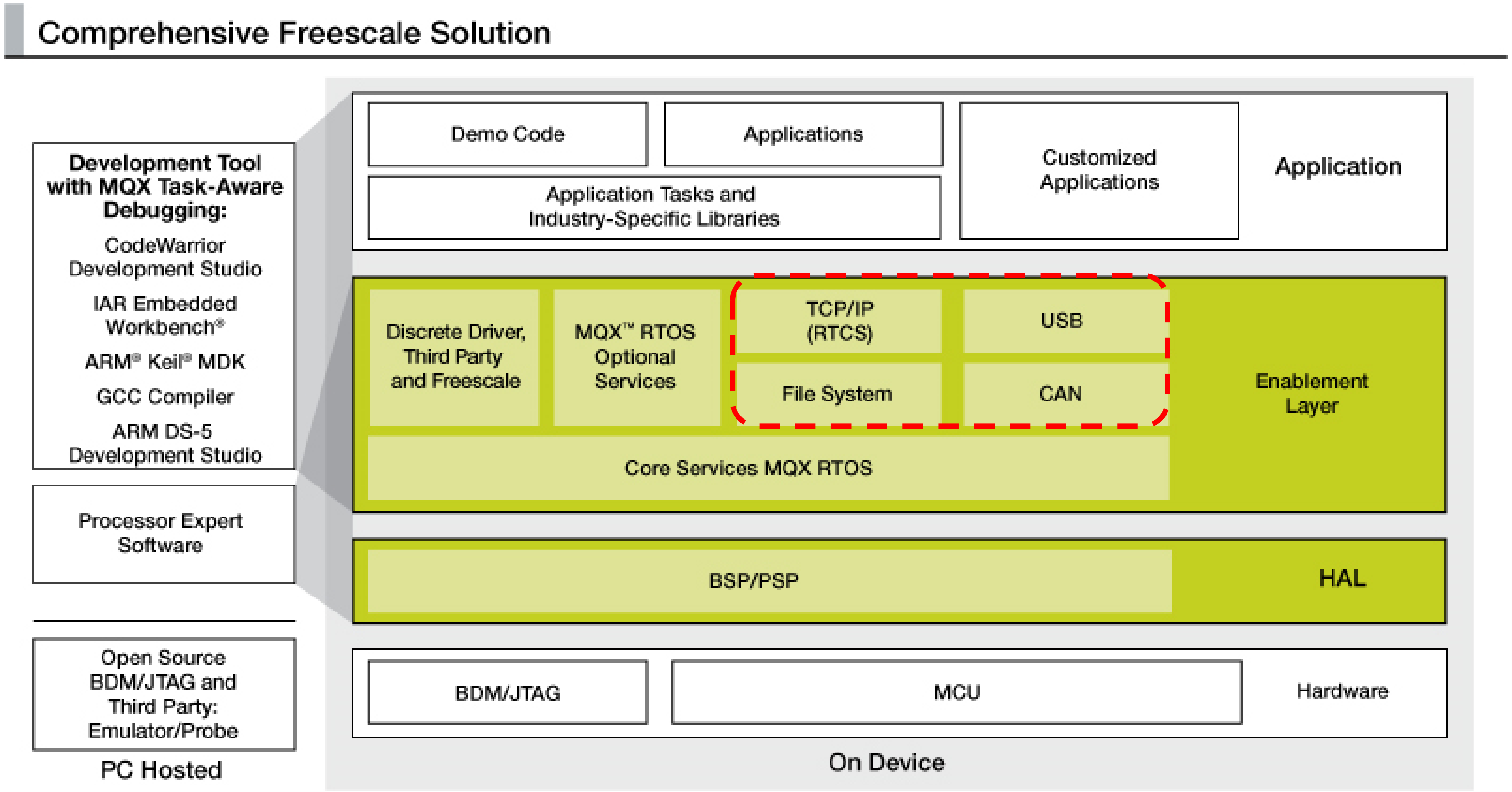
I/O Subsystems

Kernel Log

Logs

As-Needed

4.3 MQX RTOS (primer) :


Comprehensive Freescale Solution
Development Tool with MQX Task-Aware Debugging:
CodeWarrior Development Studio
IAR Embedded Workbench®
ARM® Keil® MDK
GCC Compiler
ARM DS-5 Development Studio
Processor Expert Software
Open Source BDM/JTAG and Third Party: Emulator/Probe
PC Hosted
Demo Code
Applications
Application Tasks and Industry-Specific Libraries
Customized Applications
Application
Discrete Driver, Third Party and Freescale
MQX™ RTOS Optional Services
TCP/IP (RTCS)
USB
File System
CAN
Enablement Layer
Core Services MQX RTOS
BSP/PSP
HAL
BDM/JTAG
MCU
Hardware
On Device

🟩 Freescale MQX™ Software Solutions

4.3 MQX RTOS (primer) :

Opravila („Tasks“)

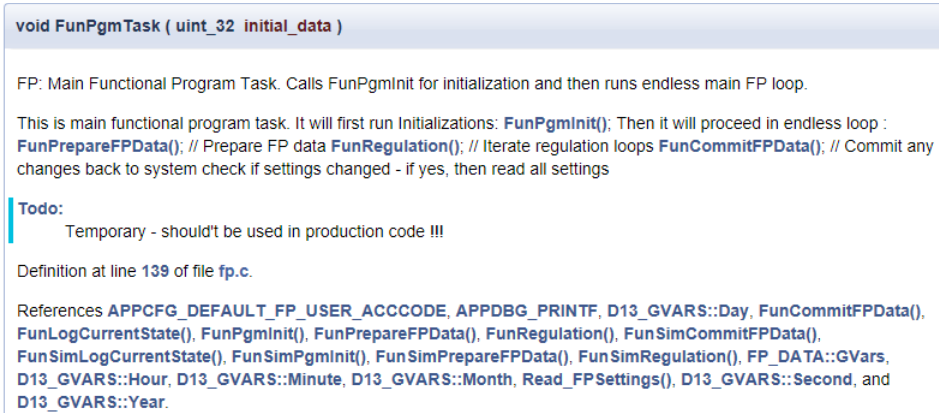
```
 const TASK_TEMPLATE_STRUCT  MQX_template_list[] =
{
   /* Task Index,     Function,          Stack,  Priority,                        Name,                          Attributes,                    Param,   Time Slice */
{ MNG_TASK,       MngTask,           1200,   TASK_PRIORITY_MNG_TASK,   MNG_TASK_DES,                  MQX_AUTO_START_TASK,         0,            0 },

{ SHELL_TASK,   ShellTask,           2000,  TASK_PRIORITY_SHELL,  SHELL_TASK_DES,      0,                            0,            0 },

{ FP_TASK,          FunPgmTask,        2000,   TASK_PRIORITY_FP,            FP_TASK_DES,                   0,                               0,            0 },

{ TNSH_TASK,      TelnetClientShell,  2000,   TASK_PRIORITY_TNETSH,TNSH_TASK_DES,     0,                            0,            0 },

{ TCPCLIENT_TASK,TCPClient_Task, 2000,  TASK_PRIORITY_TCPCLIENT,TCPCLIENT_TASK_DES,           0,                               0,            0 },

{ MODBUS_TASK,Modbus_Task,    2000,   TASK_PRIORITY_MODBUS,MODBUS_TASK_DES,  0,                            0,            0 },

{ EVTALM_TASK,EventAlmTask,       2000,   TASK_PRIORITY_EVTALM,EVTALM_TASK_DES, 0,                            0,            0 },

{ AIN_TASK,         AinTask,             500,    TASK_PRIORITY_AIN,           AIN_TASK_DES,                  0,                               0,            0 },

{ NETMNG_TASK,NetMngTask,        1000,   TASK_PRIORITY_NETMNG,NETMNG_TASK_DES,                0,                               0,            0 },

{ 0 }

};
```

4.3 MQX RTOS (primer MQX opravila) :

Glavna regulacijska zanka („FP_TASK“)

```
void FunPgmTask (uint_32 initial_data)
{
FunPgmInit();

// register task for system messages
rc = SysMsgRegister ();

// WDT control
WdtRegister (15000, WDT_ACTION_LOG);

// ------------------------------------  main execution loop -----------------------------
while (TRUE)  {

        _time_get_elapsed (&fp_start_time);   //Measure processing time fp_start_time

        WdtReset ();

        FunPrepareFPData();             // Prepare FP data
        FunRegulation();                // Iterate regulation loops
        FunCommitFPData();              // Commit any changes back to system

        _time_get_elapsed (&fp_end_time);   //Measure processing time
        _time_diff (&fp_start_time, &fp_end_time, &fp_loop_time);     // get elapsed time
        FPLoopTime=(fp_loop_time.SECONDS * 1000) + fp_loop_time.MILLISECONDS;

        _time_delay(1000-FPLoopTime);// wait for 1000 ms - loop time in ms
        }
_task_block();                                         // Shouldn't reach this point
}
```

```
/** @brief FP: Main Functional Program Task.
Calls FunPgmInit for initialization and then runs endless main FP loop.
 *
 *  This is main functional program task.
 *  It will first run Initializations: FunPgmInit();
 *  Then it will proceed in endless loop :
 *               FunPrepareFPData();     // Prepare FP data
 *               FunRegulation();        // Iterate regulation loops
 *               FunCommitFPData();      // Commit any changes back to system
 *               check if settings changed - if yes, then read all settings
*/
```



void FunPgmTask (uint_32 initial_data)

FP: Main Functional Program Task. Calls FunPgmInit for initialization and then runs endless main FP loop.

This is main functional program task. It will first run Initializations: **FunPgmInit()**; Then it will proceed in endless loop : **FunPrepareFPData()**; // Prepare FP data **FunRegulation()**; // Iterate regulation loops **FunCommitFPData()**; // Commit any changes back to system check if settings changed - if yes, then read all settings

Todo:

Temporary - should't be used in production code !!!

Definition at line 139 of file fp.c.

References **APPCFG_DEFAULT_FP_USER_ACCCODE**, **APPDBG_PRINTF**, **D13_GVARS::Day**, **FunCommitFPData()**, **FunLogCurrentState()**, **FunPgmInit()**, **FunPrepareFPData()**, **FunRegulation()**, **FunSimCommitFPData()**, **FunSimLogCurrentState()**, **FunSimPgmInit()**, **FunSimPrepareFPData()**, **FunSimRegulation()**, **FP_DATA::GVars**, **D13_GVARS::Hour**, **D13_GVARS::Minute**, **D13_GVARS::Month**, **Read_FPSettings()**, **D13_GVARS::Second**, and **D13_GVARS::Year**.