STR9 Ethernet SpeedWay

Hands-on Session 2

2007

STM EMMA Prague
Silica TMC

EasyUDP Exercise

EasyUDP Exercise

We are going to learn how to

Control a device over network

⬙ Step 1: Process UDP Frame

⬙ Step 2: See It Running

⬙ Step 3: Do it Over TCP

⬙ Step 4: See Differences Between UDP and TCP

Step 1: Process UDP Frame

Start from the original EasyWEB project.
You can find it in [exercises] EasyWEB.zip

⬦ Add UDP processing function

> **Open file**
>
> tcpip.c
>
> Insert ProcessUDPFrame function from file
>
> [exercises] mod_udp.c
>
> at the beginning of the source code

⬦ Use the function in TCP/IP process
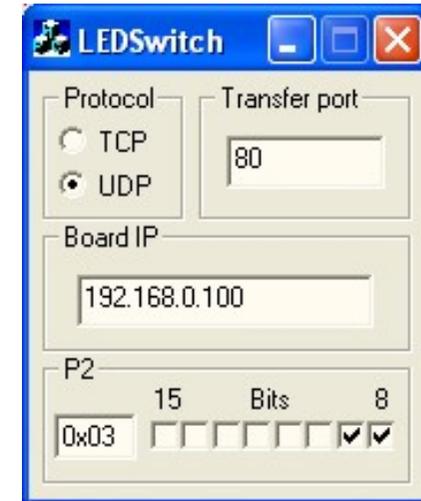
```
371    switch (ProtocolType) {
372        case PROT_ICMP : { ProcessICMPFrame(); break; }
373        case PROT_TCP  : { ProcessTCPFrame(); break; }
374        case PROT_UDP  : { ProcessUDPFrame(); break; }
375    }
```

⬦ Rebuild the project and flash the device

Step 2: See It Running

◆ Run Application LEDSwitch

You can find it in

[exercises] LEDSwitch.exe

Control the LEDs

It does not work?
What port does our
project use?

**By the way, the web
server is still working.**
Check it in web browser.

Step 3: Do it Over TCP

Shall we write TCP processing function?

Well, good luck with it. It has about 250 lines of source code not counting the auxiliary functions and it must handle timer for timeouts. Fortunately, it is already implemented.

So we will just modify the server that works above the TCP – the HTTP server.

- Add LEDs server

  Open file

  easyweb.c

  Insert LEDsServer function from file

  [exercises] mod_tcp.c

  at the beginning of the source code

- Use the function in the main loop

```
072       //HTTPServer();
073       LEDsServer();
074    }
```

- Rebuild the project and flash the device

Step 4: See Differences Between UDP and TCP
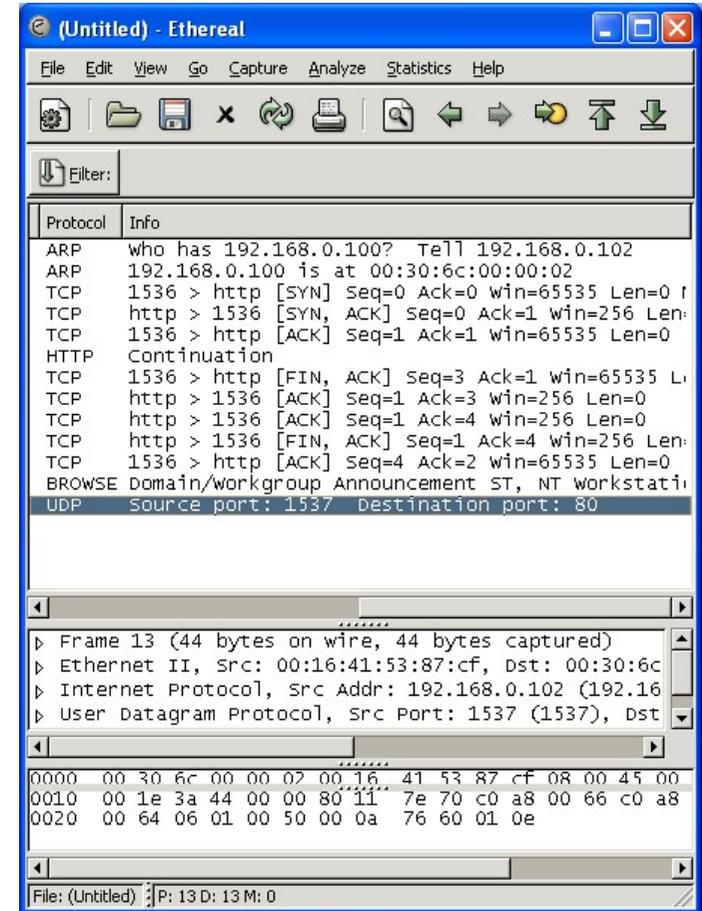
Run Application LEDSwitch and test TCP

Run Ethereal Analyzer

What to test?
- Differences between UDP and TCP traffic
    - how many packets are transmitted
    - and in which direction
- What happens (UDP vs. TCP) if you
    - change the port number
    - change the address
    - unplug the Ethernet cable
    - plug the Ethernet cable to company network

EasyMEMS Exercise

EasyMEMS Exercise

We are going to learn how to

Retrieve data from sensor using web server

- Step 1: Setup Startup Code for MEMS Sensor

- Step 2: Add MEMS Driver

- Step 3: Modify Web Page

- Step 4: Modify Web Server

- Step 5: See It Running

Step 1: Setup Startup Code for MEMS Sensor

Start from the original EasyWEB project.
You can find it in [exercises] EasyWEB.zip
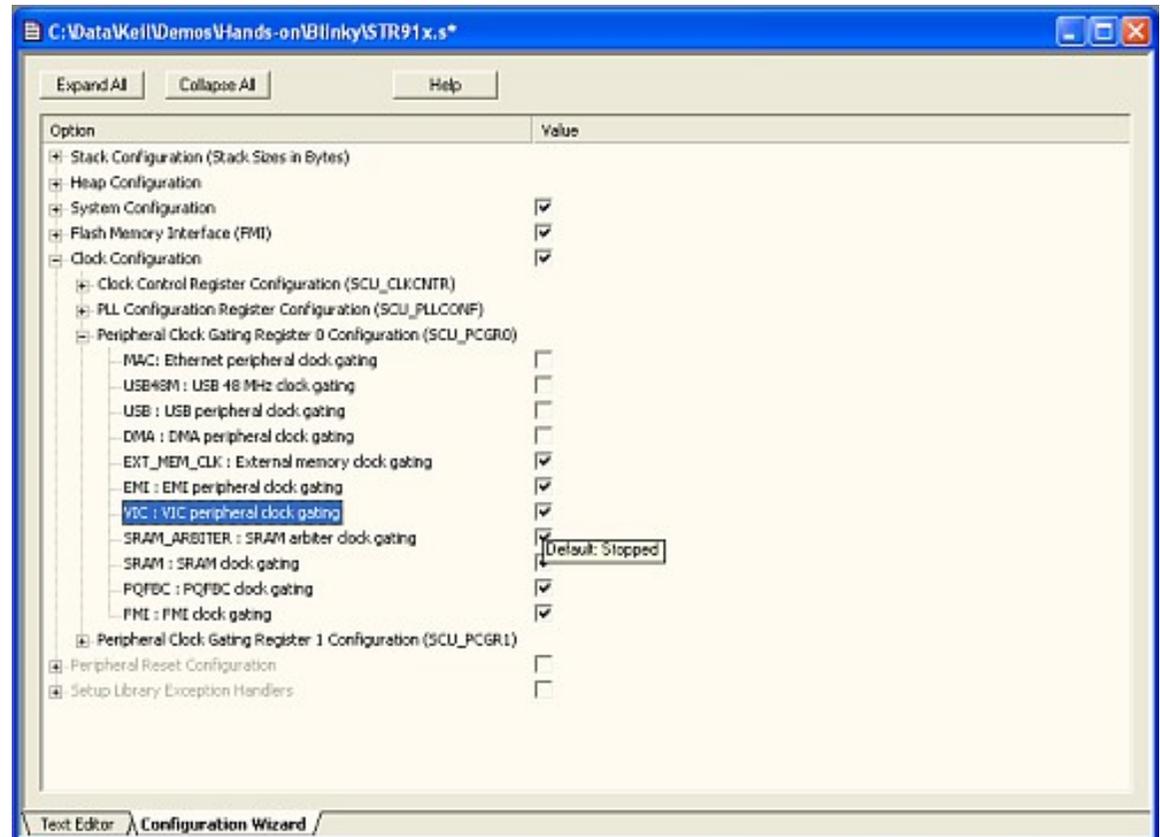
▣ Open Configuration Wizard

Project Workspace
Open File
STR91x.s

▣ Enable clocks and disable resets for peripherals related to MEMS

GPIO02
GPIO08
SSP0

C:\Data\Keil\Demos\Hands-on\Blinky\STR91x.s*

Expand All   Collapse All   Help

Option	Value
Stack Configuration (Stack Sizes in Bytes)	
Heap Configuration	
System Configuration	☑
Flash Memory Interface (FMI)	☑
Clock Configuration	☑
Clock Control Register Configuration (SCU_CLKCNTR)	
PLL Configuration Register Configuration (SCU_PLLCONF)	
Peripheral Clock Gating Register 0 Configuration (SCU_PCGR0)	
MAC: Ethernet peripheral clock gating	☐
USB48M : USB 48 MHz clock gating	☐
USB : USB peripheral clock gating	☐
DMA : DMA peripheral clock gating	☐
EXT_MEM_CLK : External memory clock gating	☑
EMI : EMI peripheral clock gating	☑
VIC : VIC peripheral clock gating	☑
SRAM_ARBITER : SRAM arbiter clock gating	☑ Default: Stopped
SRAM : SRAM clock gating	
PQFBC : PQFBC clock gating	☑
FMI : FMI clock gating	☑
Peripheral Clock Gating Register 1 Configuration (SCU_PCGR1)	
Peripheral Reset Configuration	☐
Setup Library Exception Handlers	☐

Text Editor   Configuration Wizard

Step 2: Add MEMS Driver

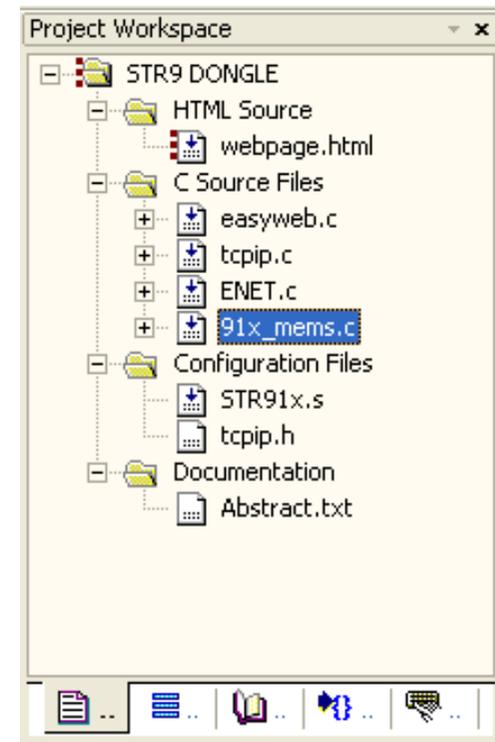▱ Copy driver to project directory

Copy files

      [exercises] 91x_mems.c

      [exercises] 91x_mems.h

▱ Add it to project

Project Workspace

Add 91x_mems.c file to group

STR9 DONGLE – C Source Files

Project Workspace

- STR9 DONGLE
  - HTML Source
    - webpage.html
  - C Source Files
    - easyweb.c
    - tcpip.c
    - ENET.c
    - 91x_mems.c
  - Configuration Files
    - STR91x.s
    - tcpip.h
  - Documentation
    - Abstract.txt

Step 3: Modify Web Page

- Open HTML source

Open file

webpage.html

- Insert HTML code that displays MEMS data

Edit page

Replace code between

```
<!------------------------------------------------------------------->
<!----------------------------- Buttons ----------------------------->
<!------------------------------------------------------------------->

   Code to be replaced
<!------------------------------------------------------------------->
<!-------------------------- END of Buttons ------------------------->
<!------------------------------------------------------------------->
```

with content of

[exercises] disp_mems.html

Check the changes you made in webpage.html using your web browser.

Do you know what are the strings _MX_ for?

They will be replaced by web server with values from MEMS sensor. We are going to modify the web server now.

Step 4: Modify Web Server

🔷 Modify web server

Edit file easyweb.c

Insert pieces of code from [exercises] mod_web.c

1

```
027  #include "webpage.h"
028  #include "91x_mems.h"
029  t_mems_data mems_data = {0,0,0,0,0,0};
```

2

```
037  SCU->GPIOTYPE[6]   = 0xOF;
038  GPIO6->DDR         = 0xOF;
039  GPIO6->DR[0xOF*4] = 0xOF;              // clear LEDs
040
041  Periph_Config_MEMs();                  // Config
042  SPI_MEMS_Init();
043
044  TCPLowLevelInit();
045
046  GPIO6->DR[0xOF*4] = 0xOB;              // Initializat
```

3

Replace functions
(from mod_web.c)
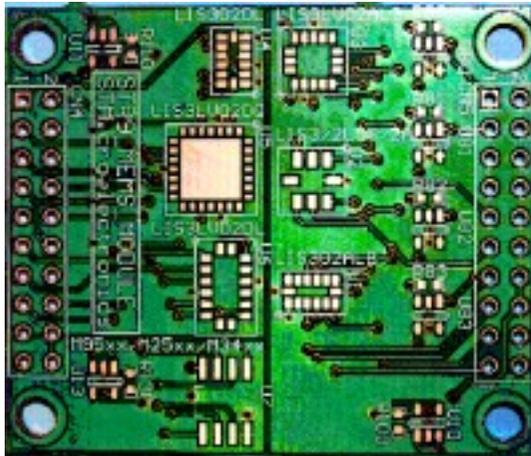
GetVal

InsertDynamicValues

Step 5: See It Running

- Rebuild project and flash the device
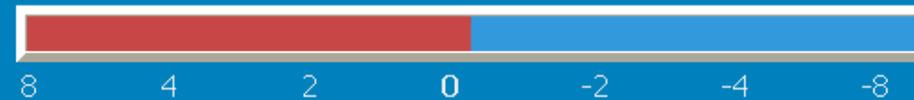- See MEMS sensor data in your browser

Are you missing something?

Power down the DONGLE and plug the **MEMS extension** board.

STR9 Ethernet Speedway

MEMS Data

MEMS X axis: (0, 26)

MEMS Y axis: (255, 235)

MEMS Z axis: (251, 252)