

Rekenaarstelsels - Computer systems - 245

[Dashboard](#) / [My courses](#) / [Rekenaarstelsels - Computer systems - 245](#) / [A2](#) / [A2](#)

Started on Saturday, 27 November 2021, 14:00

State Finished

Completed on Saturday, 27 November 2021, 16:47

Time taken 2 hours 47 mins

Information

<p>Unless the assessment instructions expressly state otherwise, the following apply:</p> <ul style="list-style-type: none"> • You are not allowed to copy this assessment or to create images of the screen showing any part of the assessment (including, but not limited to, images obtained by means of screen shots, photographs and any software applications). • This is an open-book assessment - you may consult ANY sources, either on SunLearn, Google or online notes that you have made. Notes on a portable electronic device may not be used. • You are not allowed to communicate about any aspect related to the assessment with another student or anybody else (other than the invigilator), from the time that the assessment becomes available on SUNLearn until you have submitted all your answers or until the opportunities for submitting the assessment have closed (whichever occurs last). This prohibition includes, but is not limited to, telephonic communication, SMSes, WhatsApp, Messenger, any social media, e-mail and any other form of communication. • You carry the sole responsibility for successfully submitting and uploading your completed assignment. You may not assume that submission was successful until SUNLearn has indicated expressly that your submission was accepted. No late submissions or uploads will be accepted. <p>By clicking "Next page", you acknowledge and agree to the conditions concerned. Any student who is found to have contravened these conditions will be subject to disciplinary action.</p>	<p>Tensy die instruksies vir die spesifieke assessering uitdruklik anders aandui, geld die volgende:</p> <ul style="list-style-type: none"> • Jy mag geen afskrif van hierdie assessering kopieer maak of afbeeldings van die skerm skep wat enige deel van die assessering toon nie (dit sluitskermgrepe, foto's en die gebruik van sagteware in, maar is nie daartoe beperk nie). • Dit is 'n oopboek-assessering hierdie - jy mag dus ENIGE bron gebruik, insluitend SunLearn, Google, of aanlyn notas was jy gemaak het. Notas op 'n draagbare elektroniese toestel mag nie gebruik word nie. • Jy mag met geen ander student of enige ander persoon (behalwe die toesighouer) oor enige aspek in verband met die assessering kommunikeer nie, vanaf die tyd dat die assessering op SUNLearn beskikbaar raak totdat jy al jou antwoorde ingedien het of totdat die indieningsgeleenthede vir die assessering gesluit het (wat ook al laaste gebeur). Hierdie verbod sluit in maar is nie beperk nie tot telefoniese kommunikasie, SMS'e, WhatsApp, Messenger, enige sosiale media, e-pos en enige ander vorm van kommunikasie. • Jy dra die volle verantwoordelikheid daarvoor om jou voltooië opdrag sukcesvol in te dien en te laai. Jy mag nie aanvaar dat indiening suksesvol plaasgevind het nie totdat SUNLearn uitdruklik aandui dat jou indiening aanvaar is. Opdragte wat laat ingedien word en dokumente wat laat gelaai word, sal nie aanvaar word nie. <p>Deur "Next page" te klik, neem jy kennis van en aanvaar jy die betrokke voorwaardes. Enige student wat hierdie voorwaardes verbreek, sal aan tugsappe onderworpe wees.</p>
--	---

Question 1

Complete

Marked out of 6.00

Answer true or false to the following statements regarding the STM32CubeIDE and the toolchain. Incorrect answers will earn negative marks. Select "Pass" if you do not want to supply an answer (to earn 0 marks).	Antwoord waar of vals vir die volgende stellings oor die STM32CubeIDE en die "toolchain". Verkeerde antwoorde sal negatief gemerk word. Kies die "Slaan oor" opsie as jy nie weet nie (om 'n punt van 0 te kry)
--	---

The compiler for the STM microcontroller is a cross-compiler	Die vertaler vir die STM mikrobeheerder is 'n kruis-vertaler
--	--

True/Waar



It is possible to write a program for the STM32F411VET microcontroller, that uses embedded peripherals, without making use of the graphical device configuration tool.	Dit is moontlik om 'n program vir die STM32F411VET mikrobeheerder te skryf wat van die toegewyde randapparaat gebruik, sonder om die grafiese konfigurasie venster te gebruik.
--	--

Pass/Slaan oor

true

The toolchain and the IDE is one and the same thing.	Die "toolchain" en die "IDE" is een en dieselfde ding
--	---

True/Waar

The  button from the IDE will restart a debug session	Die  knoppie van die IDE sal 'n ontfouting sessie oorbegin
--	---

False/Vals

Not important

One can set breakpoints in the disassembly view	Mens kan breekpunte in die "disassembly" venster stel
---	---

True/Waar

The best way to view the value of a local variable is using the Memory Browser window.	Die beste manier om die waarde van 'n plaaslike veranderlike te sien is deur die "Memory Browser" venster
--	---

True/Waar

Question 2

Complete

Marked out of 12.00

Consider the assembly code below. It is supposed to implement a function that receives 5 arguments (a, b, c, d, e) and then calculates the sum of all of these numbers and returns the result of the calculation and stores it at memory address 0x1000.

Due to some errors in the code it does not currently work. Please answer the questions that follow. When line numbers are asked only supply the number.

Each question is worth 2 marks.

NOTE: The Thumb instruction set is used.

Beskou die saamsteltaal-kode hieronder. Dit is veronderstel om 'n funksie te implementeer wat 5 argumente ontvang (a, b, c, d, e) en dan die som van al die getalle te bereken en die resultaat van die berekening oplewer en dan by adres 0x1000 stoor.

As gevolg van sommige foute in die kode, werk dit nie tans nie. Beantwoord asseblief die vrae wat volg. As reëlnommers gevra word, verskaf slegs die nommer.

Elke vraag is 2 punte werd.

LET WEL: Die Thumb instruksiestel word gebruik.

```

1      MOV      R5, #0x1000
2
3      MOV      R0, #1   ; a
4      MOV      R1, #2   ; b
5      MOV      R2, #3   ; c
6      MOV      R3, #4   ; d
7      MOV      R4, #5   ; e
8      BL       FUNCTION
9
10     STR      R0, [R5, #0!]
11
12     B        Done
13
14 FUNCTION
15     STMDB    SP!, {R5, R6, R7}
16     ADD     R0, R1
17     ADD     R0, R2, R3
18     ADD     R0, R3, R4
19     LDMIA   SP!, {R7, R8, R9}
20     MOV     PC, LR
21
22 DONE

```

We not gonna do assembly i think

In which line is there a syntax error (there might be more than one correct answer but only provide one answer)?

In watter lyn is daar 'n sintaksfout (daar is moontlik meer as een korrekte antwoord maar gee slegs een antwoord)?

16 maybe

Which line transgresses the AAPCS?

Watter lyn oortree die AAPCS?

☒ 7

would be 18

In which line is there a logic error?

In watter lyn is daar 'n logiese fout?

☒ 16

Which line contains memory inefficient code?

In watter lyn is daar geheue-ondoeltreffende kode?

☒ 10

Is it acceptable to write either "Done" or "DONE" when referring to the same label? (this question is marked negative)

Is dit aanvaarbaar om of "Done" of "DONE" te skryf wanneer na dieselfde etiket verwys word? (hierdie vraag word negatief gemerk)

☒ No/Nee

Is it acceptable to write either "add" or "ADD" when referring to the same instruction? (this question is marked negative)

Is dit aanvaarbaar om of "add" of "ADD" te skryf wanneer na dieselfde instruksie verwys word? (hierdie vraag word negatief gemerk)

☐ Yes/Ja

Question **3**

Incorrect

Marked out of 4.00

For this question you **must** use the provided GPIO structs such as GPIOx->ODR, GPIOx->IDR and GPIOx->BSRR. Any other approach will be penalised.

Assuming that **PB5**, **PB6**, **PB7** and **PB8** are set as outputs and **PD0** and **PD1** are set as inputs. Write some lines of code (not within a function) that matches the outputs to the inputs as follows:

- The output state of **PB5** and **PB6** should match the current input state of **PD0**.
- The output state of **PB7** and **PB8** should match the current input state of **PD1**.

Vir hierdie vraag **moet** jy die verskafde GPIO-strukture gebruik soos GPIOx->ODR, GPIOx->IDR en GPIOx->BSRR. Enige ander benadering sal gepeenaliseer word.

Aanvaar dat **PB5**, **PB6**, **PB7** en **PB8** as uitrees gestel is en **PD0** en **PD1** as intrees gestel is. Skryf 'n paar reëls kode (nie binne 'n funksie nie) wat sal verseker dat die uitrees ooreenstem met die intrees soos volg:

- Die uitree-toestand van **PB5** en **PB6** moet ooreenstem met die huidige intree-toestand van **PD0**.
- Die uitree-toestand van **PB7** en **PB8** moet ooreenstem met die huidige intree-toestand van **PD1**.

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
GPIOB->BSRR |= (((GPIOID->IDR & 0b1) << 8) + ((GPIOID->IDR & 0b1) << 7) + ((GPIOID->IDR & 0b0) << 6) + GPIOB->BSRR |= ((GPIOID->IDR & 0b0) << 5));
```

```
GPIOB->ODR = (GPIOB->ODR & 0xfffffe1f) |
              ((GPIOID->IDR & 0x1) < 6) |
              ((GPIOID->IDR & 0x1) < 7) |
              ((GPIOID->IDR & 0x2) < 8) |
              ((GPIOID->IDR & 0x2) < 9);
```

Question 4

Complete

Marked out of 5.00

The following questions are related to the differences between the BSRR and ODR registers that may be used to change the output logic level of the GPIO pins.	Die volgende vrae hou verband met die verskille tussen die BSRR- en ODR-registers wat gebruik kan word om die uittree-logikavlak van die GPIO-penne te verander.
If I set an single GPIO output pin to either 0 or 1, which of the following statements is true regarding the number of assembly instructions that are required in the most efficient implementation: (2 marks)	As ek 'n enkele GPIO-uitreepen na 0 of 1 stel, watter een van die volgende stellings is waar met betrekking tot die aantal samestelling-instruksies wat benodig word vir die doeltreffendste implementering: (2 punte)
BSRR requires more instructions than the ODR alternative / BSRR vereis meer instruksies as die ODR alternatief	
You can read from the GPIOA_BSRR register to determine the current output pin logic level. (this question is marked negative) (1 mark)	Jy kan uit die GPIOA_BSRR-register lees om die huidige logiese vlak van die uittreepen te bepaal. (hierdie vraag word negatief gemerk) (1 punt)
False / Vals	
You can read from the GPIOA_ODR register to determine the output pin logic level. (this question is marked negative) (1 mark)	Jy kan uit die GPIOA_ODR-register lees om die huidige logiese vlak van die uittreepen te bepaal. (hierdie vraag word negatief gemerk) (1 punt)
True / Waar	
You can read from the GPIOA_ODR register to determine the current logic level of an input pin. (this question is marked negative) (1 mark)	Jy kan uit die GPIOA_ODR-register lees om die huidige logiese vlak van 'n intreepen te bepaal. (hierdie vraag word negatief gemerk) (1 punt)
False / Vals	

Question 5

Complete

Marked out of 7.00

<p>In the case of the STM32F411 microcontroller, for each of the following, specify if it will be placed in Flash or SRAM memory.</p> <p>Incorrect answers will earn negative marks. Select "Pass" if you do not want to supply an answer (to earn 0 marks).</p>		<p>In die geval van die STM32F411 mikrobeheerder, vir elk van die volgende, spesifiseer of dit in Flash of SRAM geheue sal voorkom.</p> <p>Verkeerde antwoorde sal negatiewe punte verdien. Kies "Pass" as jy nie 'n antwoord wil verskaf nie (om 0 punte te verdien).</p>
Vector table / Vektor tabel	SRAM	X
Stack / Stapel	Flash	X
Interrupt handler function / Onderbrekingshanterings roetine	Flash	✓
Array declared as / Skikking verklaar as <pre>static const int array1[2] = {1,2};</pre>	Flash	X
Array declared as / Skikking verklaar as <pre>int array1[2] = {1,2};</pre>	SRAM	✓
The constant PI, as used in the following code / Die konstante PI, soos dit voorkom in die kode <pre>#define PI 3.14159 float angle = i*PI/10.0f;</pre>	SRAM	✓
A global variable / 'n Globale veranderlike	Flash	X

Question 6

Complete

Marked out of 6.00

Answer true or false to the following statements regarding microcontroller peripherals. Incorrect answers will earn negative marks. Select "Pass" if you do not want to supply an answer (to earn 0 marks).	Antwoord waar of vals vir die volgende stellings in verband met mikrobeheerder randapparaat. Verkeerde antwoorde sal negatief gemerk word. Kies "Slaan oor" as jy nie weet nie (om 'n punt van 0 te kry)
---	--

A timer/counter can be used to measure the rotation rate of a motor by counting pulses from an optical encoder.	'n Tydhouer/teller kan gebruik word om die rotasietempo van 'n motor te meet deur pulse van 'n optiese enkodeerder te tel.
---	--

True/Waar 

An ADC can be used to measure temperature from a sensor that outputs a varying amount of pulses, proportional to the temperature	'n ADC kan gebruik word om die temperatuur te meet vanaf 'n sensor wat 'n varieërende aantal pulse uitgee, proporsioneel tot die temperatuur.
--	---

True/Waar 

An SPI communications link can be used to obtain measurements from an external analog-to-digital converter that has two signals labelled SCL and SDA for communication.	'n SPI kommunikasiekanaal kan gebruik word om metings te verkry vanaf 'n eksterne analoog-na-digitaal omsetter wat twee seine, gemerk SCL en SDA, gebruik vir kommunikasie
---	--

False/Vals 

The UART peripheral is used to send and receive serial data where the other device uses two signals labelled TX and RX, with no clock signal.	Die UART randapparaat word gebruik om seriële data te stuur waar die ander toestel twee seine, gemerk TX en RX het, met geen kloksein
---	---

True/Waar 

The ADC is used to generate an analog audio signal from a MP3 sound file	'n ADC word gebruik om 'n analoog klanksein vanaf 'n MP3 klanklêer op te wek.
--	---

False/Vals 

DMA can be used to transfer data from flash memory to SRAM	DMA kan gebruik word om data oor te dra van flash geheue na SRAM
--	--

False/Vals 

Question 7

Complete

Marked out of 5.00

<p>You are using the STM32F411VET6 microcontroller with a timer clock of 16 MHz. You are tasked with generating a PWM signal in PWM Mode 2 in center-aligned mode that has a duty cycle of 70%. Given that the CCRx value is set to 21. What must the value of TIMx_ARR and TIMx_RCR be?</p>	<p>Jy gebruik die STM32F411VET6 mikrobeheerder met 'n tydhouer van 16 MHz. Jy word getaak om 'n PWM-sein in PWM-modus 2 in middel-belynde modus te genereer wat 'n dienssiklus van 70% het. Gegee dat die CCRx-waarde op 21 gestel is. Wat moet die waarde van TIMx_ARR en TIMx_RCR wees?</p>
--	---

TIMx_ARR = ☐ 70

30

TIMx_RCR = ☐ 0

not sure?

<p>For the same situation, at the transition to which value of TIMx_CNT will an update event (UEV) occur? (there might be more than one correct answer).</p>	<p>Vir dieselfde situasie, by die oorgang na watter waarde van TIMx_CNT sal 'n opdateringsgebeurtenis of "update event" (UEV) plaasvind? (daar kan moontlik meer as een korrekte antwoord wees).</p>
--	--

TIMx_CNT = ☐ 35

21

<p>For the same situation, what will the value of OCXREF be for the following TIMx_CNT values?</p>	<p>Vir dieselfde situasie, wat sal die waarde van OCXREF wees vir die volgende TIMx_CNT-waardes?</p>
--	--

When/Wanneer TIMx_CNT = 3, OCXREF = ☐ 0When/Wanneer TIMx_CNT = 25, OCXREF = ☐ 1

In the following 9 questions you will work on a program that runs on the STM32F4 emulator, with the following functionality:

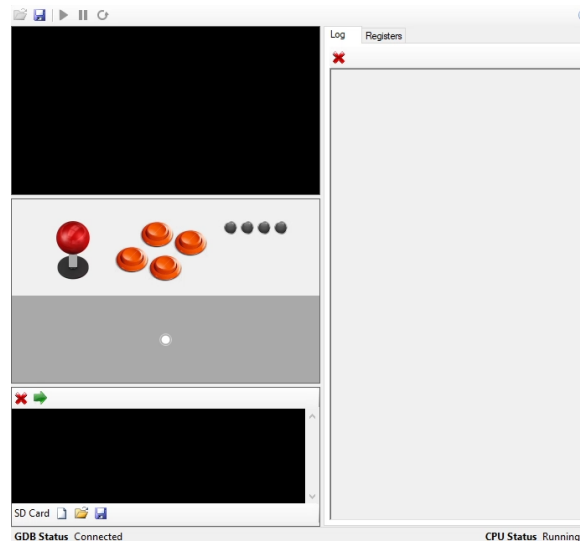
- When the user presses either of the 4 buttons (not the joystick, the 1,2,3,4 buttons) the corresponding LED should turn on. At the same time, the screen should be filled with the same colour as the LED that was activated.
- After a time interval, that depends on the button that was pressed, the LED should turn off and the screen should be filled with black.
- (If a button is pressed before the previous period expired, the LED should be set, screen filled, and timer reloaded for the new button.)

The functionality is demonstrated in the short video:

In die volgende 9 vrae gaan jy werk aan 'n program wat op die STM emuleerder uitvoer met die volgende funksionaliteit:

- Wanneer die gebruik enige van die 4 knoppies druk (nie die "joystick" nie, maar die 1,2,3,4 knoppies) moet die ooreenstemmende LED aanskakel. Tesame hiermee moet die skerm gevul word met dieselfde kleur as die LED wat geaktiveer is.
- Na 'n tyd interval wat afhang van die knoppie wat gedruk is, moet die LED weer afskakel en die skerm terugkeer na swart
- (Indien 'n knoppie gedruk word voor die vorige periode verstryk moet die LED gestel word en tydhouer herlaai word vir die nuwe knoppie)

Die funksionaliteit word gedemonstreer in die volgende kort video:

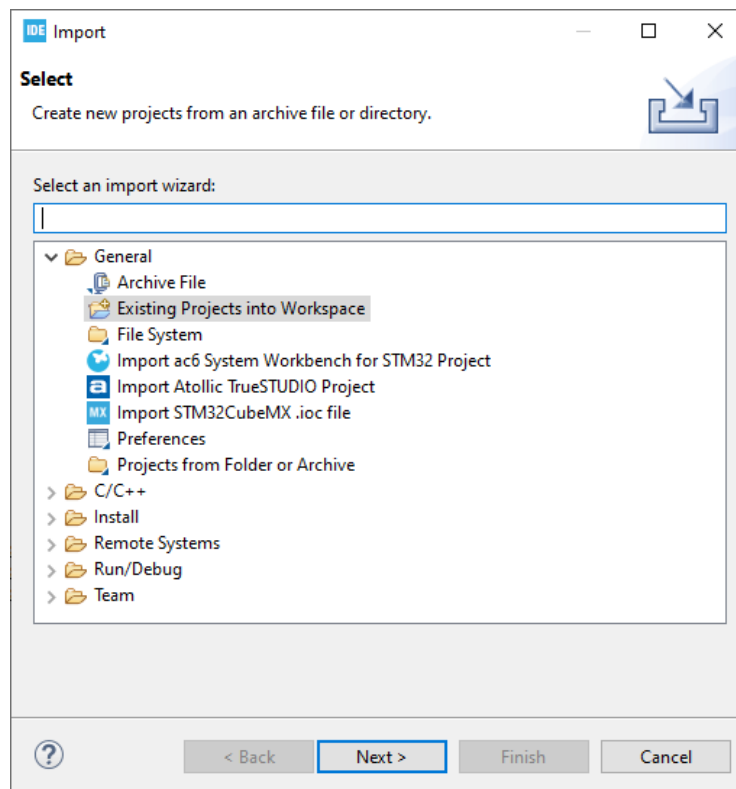


The colour to use for the screen fill, as well as the duration that each state should continue for, is summarized in the table below

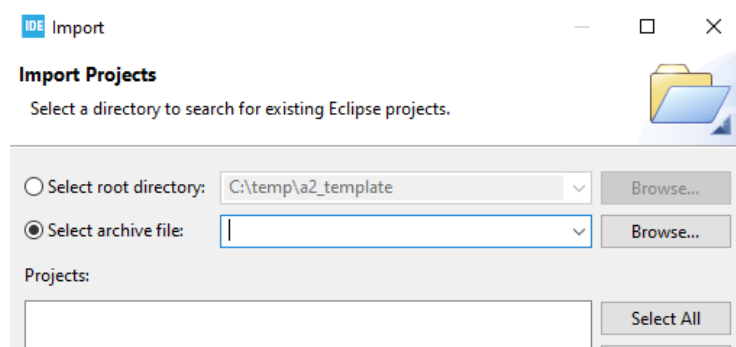
Die kleur waarmee die skerm gevul moet word, sowel as die tydsduur wat elke toestand moet duur, word in die tabel hieronder gegee:

Button	LED	Display colour index	Time active
1	1 (Green)	2	500 ms
2	2 (Orange)	42	1 s
3	3 (Red)	4	2 s
4	4 (Blue)	1	4 s

<p>You will not have to submit the BIN program file for these questions, but instead, the following questions will lead you to implement this functionality, and you will have to supply some answers and working code fragments as you go along. You should thus use the IDE and emulator to test your program before you submit answers.</p> <p>To simplify things, a project template has already been created for you. You do not have to start a new project from scratch and you should not use the device configuration tool (generate code) - all the necessary peripheral setup has already been completed in the template project.</p> <p>To make use of the supplied project file, unzip the contents of the file (A2_template.zip) on SunLearn to a temporary folder on the C: drive (do not use spaces for the folder names).</p> <p>Next, from the IDE menu, select File > Import. In the window that opens, select General > Existing Projects into Workspace, and click Next</p>	<p>Jy hoef nie die BIN lêer vir jou program op te laai nie. Maar die volgende reeks vrae sal jou lei om die funksionaliteit te implementeer en jy sal kode stukke en ander antwoorde moet voltooi soos jy aangaan. Jy moet dus die emuleerder in IDE gebruik om jou antwoorde te toets voor jy dit indien.</p> <p>Om hierdie vrae te vergemaklik, is daar 'n projek templaar beskikbaar gemaak op SunLearn. Jy hoef nie 'n nuwe projek van voor af te skep nie en jy moet ook nie probeer om die konfigurasie funksie van die IDE (generate code) te gebruik nie. Al die nodige randapparaat opstelling is reeds voltooi in die projek templaar.</p> <p>Om gebruik te maak van hierdie templaar projek, moet jy die inhoud van die A2_template.zip lêer op SunLearn "unzip" na 'n tydelike plek op die C: dryf (maak seker om nie spasies te gebruik vir "folder" name nie)</p> <p>Hierna moet jy die projek invoer (import). In die IDE spyskaart, kies die File > Import opsie. In die venster wat verskyn, kies General > Existing Projects into Workspace en kies Next.</p>
--	---



<p>In the following window, click the Browse button (for the Select root directory option), and navigate to the folder with your project.</p>	<p>In die volgende venster, kies die Browse opsie (vir die "Select root directory" opsie), en navigeer na die ligging op die hardeskyf waar jy die projek geplaas het.</p>
---	--



Under Projects, the name of the project to import will appear with a checkbox (led_display_timer). Make sure it is checked, then click Finish. The project will now appear in the project explorer, and you can build and debug it.

Note that the program won't do anything - it simply contains all the peripheral setup. For completeness, the following peripheral setup is performed in the template:

- PD12, PD13, PD14 and PD15 was selected as GPIO_Output
- PA0, PA1, PA2 and PA3 was selected as GPIO_EXTIx. Interrupts for all 4 these signals were enabled and trigger only on the rising edge.
- TIM2 was configured with a clock source of "Internal Clock". The APB peripheral clock for the timers is 16MHz (the default), and a prescaler value of 15999 was specified for TIM2. The TIM2 global interrupt was enabled.
- A memory-to-memory DMA stream (DMA2 Stream 0) was added, with a transfer width of 1 word for both the source and destination. The destination address was enabled to auto-increment but the source address was not.

You will also see that, for convenience, there is a function in the template project in main.c, called *setstate*. This high-level function will

- call the *setleds* function to enable/disable the LEDs based on the current state
- call the *fillscreen* function to fill the screen based on the current state
- Activate or deactivate the timer.

The implementation for the sub-functions (*setleds*, *fillscreen*, *starttimer* and *stoptimer*) is empty and you will complete these in the questions that follow.

Die naam van die projek (led_display_timer) sal onder "Projects" vertoon met 'n "checkbox" langs. Maak seker die checkbox is gekies en kies dan Finish. Die projek sal nou in die IDE se "Project Explorer" venster vertoon en jy behoort dit te kan bou en toets.

Let wel dat die program niks gaan doen nie - dit bevat bloot die nodige randapparaat opstelling. Vir volledigheid word hierdie opstelling hieronder gelys:

- PD12, PD13, PD14 en PD15 is opgestel as GPIO uitree
- PA0, PA1, PA2 en PA3 is opgestel as eksterne onderbreking (GPIO_EXTIx). Onderbrekings vir al 4 se is geaktiveer en sal op die stygende vlak sneller
- TIM2 is gekonfigureer deur die interne klok te kies as bron. Die APB randapparaat klok vir die tellers is 16MHz (die versuimwaarde) en 'n vooraf skaleringswaarde van 15999 was gekies. Die TIM2 globale onderbreking is geaktiveer.
- 'n Geheue-na-geheue DMA kanaal (DMA2 Stream 0) is bygevoeg met 'n oordragwydte van 1 woord vir beide die bron en bestemming. Die bestemming adres is gekies om vanself te inkrementeer tydens die oordrag, maar die bron adres nie.

Jy sal ook oplet dat daar 'n funksie in die templaet geskep is (in main.c) met die naam *setstate*. Hierdie hoëvlak funksie sal die volgende bewerkstelling

- roep die *setleds* funksie om die LEDs te aktiveer of af te skakel afhangend van die huidige toestand
- roep die *fillscreen* funksie om die skerm te vul met 'n konstante kleur afhangend van die huidige toestand
- Aktiveer of deaktiveer die tydhouer.

Die implementering van die sub-funksies (*setleds*, *fillscreen*, *starttimer* en *stoptimer*) is leeg en jy sal dit voltooi in die vrae wat volg.

Question 8

Complete

Marked out of 2.00

<p>Given the peripheral setup that was explained previously for the A2_template project, what will the values of the following GPIO registers be? (Only the pins that were mentioned were configured. The other will remain at their default state)</p> <p>Specify all eight digits of the 32-bit hex values</p>	<p>Gegee die randapparaat opstelling wat in die vorige vraag genoem is vir die A2_template projek, wat sal die waardes van die volgende GPIO registers wees? (Slegs die penne wat vroer genoem is is opgestel. Die res sal by hul versuim toestand wees)</p> <p>Skryf al agt syfers van die 32-bis heksadesimale waardes</p>
--	--

GPIOA_MODER: 0x 0C00 0000

think 0x00000000

GPIOB_MODER: 0x 0000 0000

0x55000000

Question 9

Complete

Marked out of 3.00

<p>The first functionality that we will add to the template, is to turn an LED on. We are not going to use the HAL framework for this, but rather write directly to the relevant GPIO output register. To improve our code readability we will make use of a #define compiler directive to assign a "friendly" name to the address of the GPIO output register. The definition should be such that we can write code statements like:</p>	<p>Die eerste funksionaliteit wat ons sal toevoeg tot die templaas is om 'n LED aan te skakel. Ons gaan nie die HAL funksies daarvoor gebruik nie, maar eerder direk na die relevant GPIO registers skryf. Om ons kode meer leesbaar te maak, maak ons gebruik van 'n #define vertaler aanwysing om 'n "vriendelike" naam te gee vir die adres van die GPIO uittree register. Die definisie moet so geskryf word sodat ons kode soos hierdie kan skryf:</p>
---	---

```
uint32_t ledstate = GPIOB_ODR;
GPIOB_ODR = 0x1000; // switch on PD12
```

Complete the #define statement: / Voltooi die #define stelling:

#define GPIOB_ODR & ((uint32_t) 0x 1000)

0x0x4002 0C14

Question **10**

Incorrect

Marked out of 4.00

Next, you have to implement the *setleds* function. It has the following prototype:

```
void setleds(int state);
```

The function implementation should switch on the LED corresponding to the state variable. Possible values for *state* ranges from 0 to 4. If *state* equals 0, all LEDs should be off. If *state* equals 1, LED1 should turn on while all the other LEDs are turned off, and so on for *state* equal to 2, 3, and 4.

The function implementation has to make use of the `GPIO_ODR` definition from earlier. You may not use the HAL functions for this. You also may not use the `BSRR` register. Your implementation has to preserve the state of the other GPIO pins for port D.

You may verify the implementation of your *setleds* by calling the *setstate* function (which will call *setleds*) from the main loop. You may use the following test code in the main loop:

```
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
  setstate(1);
  HAL_Delay(500);
  setstate(2);
  HAL_Delay(500);
  setstate(3);
  HAL_Delay(500);
  setstate(4);
  HAL_Delay(500);
  setstate(0);
  HAL_Delay(500);
}
/* USER CODE END WHILE */
```

```
/* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
```

Volgende moet jy die *setleds* funksie implementeer. Dit het die volgende prototipe:

```
void setleds(int state);
```

Die funksie implementering moet die LEDs aanskakel in ooreenstemming met die state parameter. Die state parameter kan waardes van 0 tot 4 aanneem. As *state* gelyk aan 0 gestel is, moet alle LEDs afgeskakel word. As dit 'n waarde van 1 het, moet LED1 aangeskakel word terwyl al die ander LEDs afgeskakel word, en so aan vir *state* gelyk aan 2, 3, en 4.

Die funksie implementering moet gebruik maak van die `GPIO_ODR` definisie van die vorige vraag. Jy mag nie HAL funksies hiervoor gebruik nie. Jy mag ook nie die `BSRR` register gebruik nie. Jou implementering moet die toestand van die ander GPIO penne vir poort D behou.

Jy kan jou implementering toets deur die *setstate* funksie (wat *setleds* sal roep) in jou hooflus te roep. Jy mag die volgende toets kode in die hooflus gebruik:

This will cycle through all the states, with a delay of 5s between states.

Once you have tested your implementation, copy the entire function including the function header into the answer box below.

IMPORTANT: Do not attempt to redefine `GPIO_ODR` in the answer box. It has already been defined outside this marking template.

Hierdie kode sal veroorsaak dat deur al die toestande gestap word met 'n 5s vertraging tussen toestande.

Nadat jy jou kode getoets het, kopieer die volledige funksie insluitend die funksie naam in die antwoord spasie hieronder.

BELANGRIK! Moet nie `GPIO_ODR` herdefinieer in jou antwoord nie. Dit is alreeds gedefinieer in die merk templaar.

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
void setleds(int state)
{
    if (state == 0) {
        GPIOD->ODR &= ~(((~GPIOA->IDR) & 0b1) << 15) + ((~(GPIOA->IDR) & 0b10) << 13)
        + ((~(GPIOA->IDR) & 0b100) << 11) + ((~(GPIOA->IDR) & 0b1000)
        << 9));
    }
    else if (state == 1) {
        GPIOD->ODR |= (GPIOA->IDR & 0b1) << 15;
    }
    else if (state == 2) {
        GPIOD->ODR |= (GPIOA->IDR & 0b1) << 13;
    }
    else if (state == 3) {
        GPIOD->ODR |= (GPIOA->IDR & 0b1) << 11;
    }
    else {
```

Because of prev def. only
GPIOD_ODR needed. Not
the struct implement used
here...

state == 1
GPIOD_ODR = (GPIOD_ODR & 0xffffefff) | 0x1;
etc.

Question 11

Incorrect

Marked out of 5.00

<p>Next, you have to implement the <i>fillscreen</i> function. This function also takes a single parameter - the display state.</p> <pre>void fillscreen(int state);</pre> <p>Depending on the <i>state</i> parameter, the screen has to be filled to the colour specified in the table earlier.</p> <p>You have to use DMA memory-to-memory transfer for this. Remember that the DMA setup was done so that an entire word (32-bits) is transferred from the source to the destination. The source address does not increment, so you can specify an address where the constant colour to use for the fill procedure is stored. A global variable called <i>src_fill</i> is already defined for this purpose.</p> <p>The destination should be the start of the screen, and the number of elements to transfer should equal all the pixels on the screen divided by four. You have to use the <i>HAL_DMA_Start_IT</i> function to initiate the transfer. (It does not matter that this function will cause an interrupt - we won't make use of it).</p> <p>Complete the function implementation to</p> <ul style="list-style-type: none"> fill the global <i>src_fill</i> variable with the correct value depending on the state argument Initiate the DMA transfer. <p>As before, test your implementation using the emulator by calling the <i>fillscreen</i> function in the main function with different arguments. Note that you have to place the call to <i>fillscreen</i> after the peripheral initialisation has been done. The same test code that was placed in the main loop can be used to verify the working of your function. The screen colour should now change to match with the LEDs.</p> <p>If you are satisfied with your implementation, copy the <i>fillscreen</i> function implementation in the answer box. Do not attempt to redefine the <i>scr_fill</i> or <i>hdma_memtomem_dma2_stream0</i> global variables. It has already been defined outside this marking template.</p>	<p>Volgende moet jy die <i>fillscreen</i> funksie implementeer. Hierdie funksie het ook 'n enkele parameter - die toestand om te vertoon.</p> <pre>void fillscreen(int state);</pre> <p>Afhangende van die <i>state</i> parameter moet die skerm gevul word met die kleur wat in die voorafgaande tabel aangedui is.</p> <p>Jy moet DMA geheue-na-geuehe oordrag gebruik hiervoor. Onthou dat die DMA opstelling reeds gedoen is sodat 'n volledige woord (32-bisse) op 'n slag vanaf die bron na die bestemming oorgedra word. Die bron adres inkrementeer nie, dus kan jy 'n adres spesifiseer waar die konstante kleur wat vir die vul prosedure gebruik moet word gestoor is. 'n Globale veranderlike met die naam <i>src_fill</i> is alreeds verklaar vir hierdie doel.</p> <p>Die bestemming moet die begin van die skerm geheue wees en die aantal elemente om oor te dra moet gelyk wees aan al die spikkels op die skerm gedeel deur 4. Jy moet die <i>HAL_DMA_Start_IT</i> funksie gebruik om die oordrag te inisieer. (Dit maak nie saak dat hierdie funksie 'n onderbreking veroorsaak nie - ons gaan nie van die onderbreking gebruik maak nie),</p> <p>Voltooi die funksie se implementering deur:</p> <ul style="list-style-type: none"> stel die globale <i>src_fill</i> veranderlike met die korrekte waarde afhangend van die state parameter Inisieer die DMA oordrag <p>Soos vantevore kan jy jou implementering toets met die emuleerder, deur die <i>fillscreen</i> funksie te roep vanaf die main funksie met verskillende argumente. Let wel dat jy die <i>fillscreen</i> funksie-roep moet plaas na die randapparaat inisialisering voltooi is. Dieselfde toets kode wat in die hooflus in vorige vraag bygevoeg is kan gebruik word om jou funksie te toets. Die skerm kleur moet nou verander om te pas by die LED wat aangeskakel is.</p> <p>As jy klaar getoets het, kopieer die <i>fillscreen</i> funksie implementering in die antwoord spasie. Moet nie die <i>scr_fill</i> of <i>hdma_memtomem_dma2_stream0</i> globale veranderlike weer definieer in die antwoord hieronder nie.</p>
--	---

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

Again, GPIOD_ODR...

Still missing
HAL_DMA_Start_IT function
to initiate the DMA.
Also, i think he messed up
the scr_fill parts

```
void fillscreen(int state)
{
    if (state == 0){
        GPIOD->ODR &= ~(((~GPIOA->IDR) & 0b1) << 15) + ((~GPIOA->IDR) & 0b10) <<
13)
+ ((~GPIOA->IDR) & 0b100) << 11) + ((~GPIOA->IDR) & 0b1000) << 9));
        scr_fill(0);
    }
    else if (state == 1) {
        GPIOD->ODR |= (((GPIOA->IDR & 0b1) << 15);
        scr_fill(1);
    }
    else if (state == 2) {
        GPIOD->ODR |= ((GPIOA->IDR & 0b10) << 13);
        scr_fill(2);
    }
    else if (state == 3) {
```

Question 12

Complete

Marked out of 4.00

The DMA implementation used word transfers (32 bits, or 4 bytes at a time). Which of the following combinations will fill the entire screen?

Die DMA implementering gebruik "word" lengte oordrag (32 bisse, of 4 grepe op 'n slag). Watter van die volgende kombinasies sal ook die hele skerm vul?

- ☐ Data Width:byte, Transfer length:16000 / Data wydte: greep, Oordrag lengte: 16000
- ☐ Data Width:byte, Transfer length:32000 / Data wydte: greep, Oordrag lengte: 32000
- ☒ ✓Data Width:byte, Transfer length:64000 / Data wydte: greep, Oordrag lengte: 64000
- ☐ Data Width:half word, Transfer length:16000 / Data wydte: half-woord, Oordrag lengte: 16000
- ☒ ✓ Data Width:half word, Transfer length:32000 / Data wydte: half-woord, Oordrag lengte: 32000
- ☐ Data Width:half word, Transfer length:64000 / Data wydte: half-woord, Oordrag lengte: 64000
- ☒ 64000 ✓Data Width:word, Transfer length:16000 / Data wydte: woord, Oordrag lengte: 16000
- ☐ Data Width:word, Transfer length:32000 / Data wydte: woord, Oordrag lengte: 32000
- ☐ Data Width:word, Transfer length:64000 / Data wydte: woord, Oordrag lengte: 64000

we didnt work with
the screen

What is the benefit to transferring a word at a time in this case?

Wat is die voordeel om die oordrag met woord elemente te doen?


The transfer takes fewer clock cycles to complete / Die oordrag neem minder kloksiklusse om te voltooi

Question **13**

Complete

Marked out of 1.00

<p>We will now implement the timer functionality to automatically go back to state 0 (all LEDs off, and black screen) after the time expired.</p> <p>Before we write the code for this, given the prescale value that was specified earlier, what is the time interval between increments of the counter register? (answer in ms units)</p>	<p>Ons gaan nou die tydhouer funksie gebruik om outomaties terug te keer na toestand 0 (alle LEDs afgeskakel met 'n swart skerm) na die tydhouer verstryk het.</p> <p>Voordat ons kode hiervoor skryf: gegee die vooraf skalering waarde wat vroeer genoem is, wat is die tyd interval tussen inkremte van die teller register? (antwoord in ms eenhede)</p>
---	--

 Answer: 500 

Question **14**

Correct

Marked out of 1.00

<p>The manner in which the timer functionality will be implemented is as follows: if the user buttons are pressed, the <i>setstate</i> function will be called (we will do this in the last step). The <i>setstate</i> function will then activate the TIM2 timer module to count up to the reload value. An interrupt will get generated and in the interrupt handler we will set the state to 0 by calling <i>setstate</i> from the timer interrupt handler.</p> <p>The last part is easy: we simply add a line</p> <pre>setstate(0);</pre> <p>to the timer interrupt handler in stm32f4xx_it.c.</p> <p>After making this change, copy your timer interrupt handler function into the answer box below</p>	<p>Die manier waarop die tydhouer funksionaliteit geïmplementeer moet word is as volg:</p> <p>Indien die gebruiker een van die knoppies druk moet die <i>setstate</i> funksie geroep word. (ons sal dit in die laaste stap doen). Die <i>setstate</i> funksie sal dan die TIM2 tydhouer module aktiveer om te tel tot by die herlaai waarde. 'n Onderbreking sal gegenereer word en in die tydhouer se onderbrekingsfunksie word die toestand na 0 gestel deur die <i>setstate</i> funksie te roep.</p> <p>Hier laaste deel behels bloot om hierdie lyn by die tydhouer onderbrekingsfunksie in stm32f4xx_it.c by te voeg:</p> <pre>setstate(0);</pre> <p>Nadat jy hierdie verandering gemaak het, kopieer jou tydhouer onderbrekingsfunksie in die antwoord spasie hieronder.</p>
--	--

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
void TIM2_IRQHandler(void)
{
    /* USER CODE BEGIN TIM2_IRQn 0 */
    setstate(0);
    /* USER CODE END TIM2_IRQn 0 */
    HAL_TIM_IRQHandler(&htim2);
    /* USER CODE BEGIN TIM2_IRQn 1 */

    /* USER CODE END TIM2_IRQn 1 */
}
```

Question **15**

Incorrect

Marked out of 5.00

<p>The next step is to implement the <i>starttimer</i> and <i>stoptimer</i> functions. For this, you will have to use the following funtions:</p> <ul style="list-style-type: none"> • HAL_StatusTypeDef HAL_TIM_Base_Start_IT(TIM_HandleTypeDef *htim); • HAL_StatusTypeDef HAL_TIM_Base_Stop_IT(TIM_HandleTypeDef *htim); 	<p>Die volgende stap is om die <i>starttimer</i> en <i>stoptimer</i> funksies te implementeer. Hiervoor moet jy die volgende funksies gebruik:</p>
<p>In the <i>starttimer</i> function, you have to enable the timer 2 module, and enable the interrupt for it. But prior to doing this, you have to change the reload value. This is accomplished by directly setting the ARR register, using the following line of code:</p> <pre>htim2.Instance->ARR = ???</pre> <p>Note that you have to set the reload value dependent on the <i>state</i> parameter.</p> <p>In the <i>stoptimer</i> function, you simply have to disable the timer.</p> <p>You can still use the test code from previously (that cycles though the different states every 5s). With your new modifications with timer functionality, the state should still change every 5s, but the state should automatically go to state 0 (no LEDs active and black screen) after the required time interval.</p> <p>After you have verified your implementations, copy the code for the <i>starttimer</i> and <i>stoptimer</i> functions into the answer box.</p>	<p>In die <i>starttimer</i> funksie moet jy die tydhouer 2 module aktiveer en die onderbreking aktiveer. Maar voor jy dit doen moet die herlaai waarde verander. Dit word bewerkstellig deur direk die ARR register te verstel, met die volgende lyn kode:</p> <pre>htim2.Instance->ARR = ???</pre> <p>Let dat jy die herlaai waarde moet stel in ooreenstemming met die <i>state</i> parameter.</p> <p>In die <i>stoptimer</i> funksie moet jy die teller weer deaktiveer.</p> <p>Jy kan steeds jou toetskode van die vorige vraag (wat deur die verskillende toestande stap elke 5s). Met jou nuwe veranderinge van hierdie vraag sal die toestand nogsteeds elke 5s verander, maar die toestand moet outomaties terugkeer na 0 (geen LEDs aangeskakel en swart skerm) na die vereiste tydinterval.</p> <p>Nadat jy jou kode getoets het, kopieer die kode vir die <i>starttimer</i> en <i>stoptimer</i> funksies in die antwoord spasie.</p>

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
void starttimer(int state)
{

}

void stoptimer()
{

}
```

Question **16**

Incorrect

Marked out of 2.00

<p>In this final question for the emulator program, you have to write the code that will cause the state to change based on button inputs. This is done by simply calling the <i>setstate</i> function from the EXTI interrupt handlers. EXTI0 should call <i>setstate(1)</i>, EXTI1 should call <i>setstate(2)</i> etc.</p> <p>Prior to testing your code, remove all the lines of test code from the main loop. (The main loop should not execute any code).</p> <p>After verifying, copy all of your EXTI interrupt handler functions from stm32f4xx_it.c into the answer box.</p>	<p>In hierdie laaste vraag van die emuleerder program moet jy kode skryf wat sal veroorsaak dat die toestand verander gebasseer op die knoppie intree. Dit word gedoen deur die <i>setstate</i> funksie te roep vanaf die EXTI onderbrekings-hanteringfunksies. EXTI0 moet <i>setstate(1)</i> roep, EXTI1 moet <i>setstate(2)</i> roep, en so aan.</p> <p>Voor jy jou kode toets, maak seker om al die vorige toetskode uit die hooflys te verwyder. (Die hooflus is nie veronderstel om enige kode uit te voer nie)</p> <p>Nadat jy jou kode getoets het, kopieer al die EXTI onderbrekings funksies van jou stm32f4xx_it.c lêer in die antwoord spasie hieronder.</p>
---	---

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```

/**
 * @brief This function handles EXTI line0 interrupt.
 */
void EXTI0_IRQHandler(void)
{
    /* USER CODE BEGIN EXTI0_IRQn 0 */
    setstate(0)
    /* USER CODE END EXTI0_IRQn 0 */
    HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_0);
    /* USER CODE BEGIN EXTI0_IRQn 1 */
    setstate(1)
    /* USER CODE END EXTI0_IRQn 1 */
}

/**
 * @brief This function handles EXTI line1 interrupt.
 */
void EXTI1_IRQHandler(void)

```

Question 17

Complete

Marked out of 8.00

A smart power meter connects to a mains power line and measures the current that flows. The device has the following requirements:

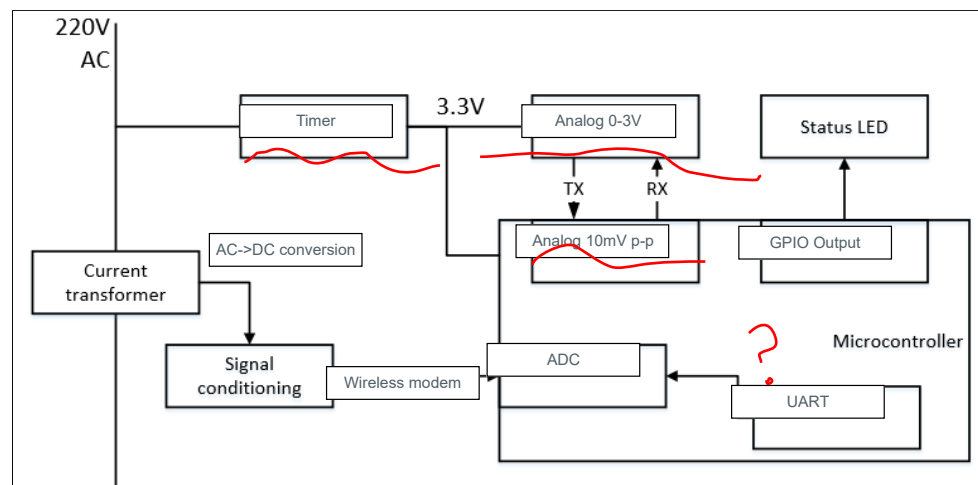
1. It has to make use of the STM32F411VE microcontroller
2. The 220V AC current measurement is performed using a Current Transformer (CT), and produces a maximum -10mV to 10mV peak-to-peak signal to correspond to a +/- 13A current on the 220V side.
3. The power meter should sample the signal from the Current Transformer and determine the RMS (Root-Mean-Square) value of the AC signal.
4. The Current Transformer signal should be sampled at 1000Hz.
5. Sampling of the CT signal should occur in the Interrupt Handling function of a timer
6. The device should be powered from the mains power supply.
7. The measured current (RMS value) should be communicated to a server using a wireless interface. For this, the power meter will make use of a wireless modem with a UART interface.
8. The microcontroller and modem require a 3.3V power supply.
9. A flashing LED should be used to indicate the status of the power meter.

Complete the block diagram of the system by placing the labels in the correct positions.

'n Slim kragmeter verbind aan 'n huis se 220V kragtoevoer en meet die stroom wat vloei. Die toestel het die volgende ontwerpvereistes:

1. Dit moet gebruik maak van 'n STM32F411VE mikrobeheerder
2. Die 220V wisselstroom meting word verrig deur 'n Stroom Transformator (CT) en lewer 'n maximum -10mV tot 10mV piek-tot-piek sien om ooreen te stem met 'n -13A na +13A stroom aan die 220V kant van die transformator.
3. Die kragmeter moet die sein vanaf die Stroom Transformator monster en vandaar die RMS (wortel-gemiddeld-kwadraat) waarde bepaal van die wisselstroom wat vloei.
4. Die stroomlesing moet teen 1000Hz gemonster word
5. Die monster proses van die stroomlesing moet uitvoer binne 'n onderbrekings roetine van 'n tydhouer
6. Die toestel moet krag verkry vanaf die 220V huistoevoer.
7. Die gemonsterde RMS stroomlesing moet aan 'n bediener gekommunikeer word deur middel van 'n draadlose koppelvlak. Hiervoor sal die kragmeter gebruik maak van 'n modem met 'n UART koppelvlak na die mikrobeheerder.
8. Die mikrobeheerder en modem vereis 'n 3.3V spanningsbron.
9. 'n flitsende LED moet die status van die toestel aandui.

Voltooi die blokdiagram van die stelsel deur die kentekens op die regte plekke te plaas.



Jump to...

A2_template_new ▶