Type A Board Dev Guide I

Wilson 2020.7.19

https://github.com/TAMU-Robomasters/Tutorial

Roadmap

1. STM32CubeMX, Keil uvision
2. LED, GPIO
3. Timer
4. PWM, passive buzzer, servo
5. Buttons
6. USB
7. Flash
8. I2C, IST8310 (magnetic sensor)
9. OLED
10. BMI088 (gyroscope)
11. Motor control with CAN
12. freeRTOS
13. IMU
14. Chassis tasks
15. Gimbal control
16. BIG PICTURE

Software & Workflow:

STM32CubeMX and Keil uvision 5 are needed for this series of tutorial.

Workflow:

1. Start a STM32CubeMX project
2. Configure
3. Use STM32CubeMX to generate code using templates, i.e. the default code structure and libraries
4. Open and edit the code in Keil uvision
5. Generate binary file
6. Connect the computer to the dev board using JLink, in SWD mode (using the SWD pin layout)
7. Download binary file to the dev board
8. Good to go

Step 1:

Launch CubeMX

File -> New Project

Search stm32f407ig

->STM32F407IGHx

After the right MCU being selected, click “Start Project”

Step 2:

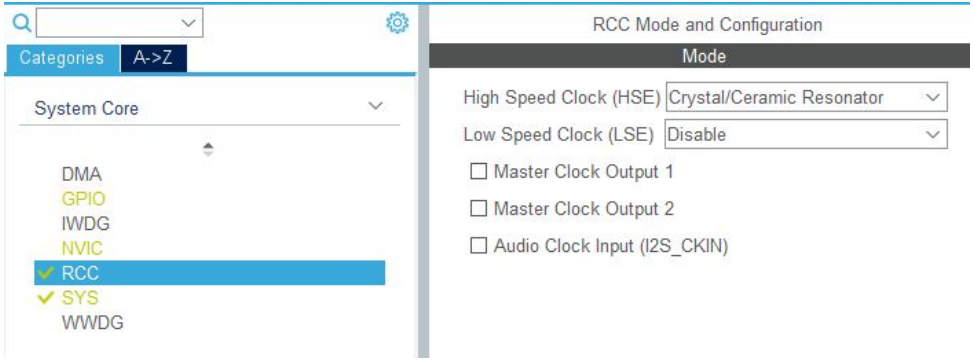
On the left pane:

Open System Core

-> RCC

In RCC Mode and Configuration:

Open High Speed Clock (HSE), select Crystal/Ceramic Resonator



Step 3:

Clock Configuration

Input frequency = 12

Select HSE

/M = /6

*N = X 168

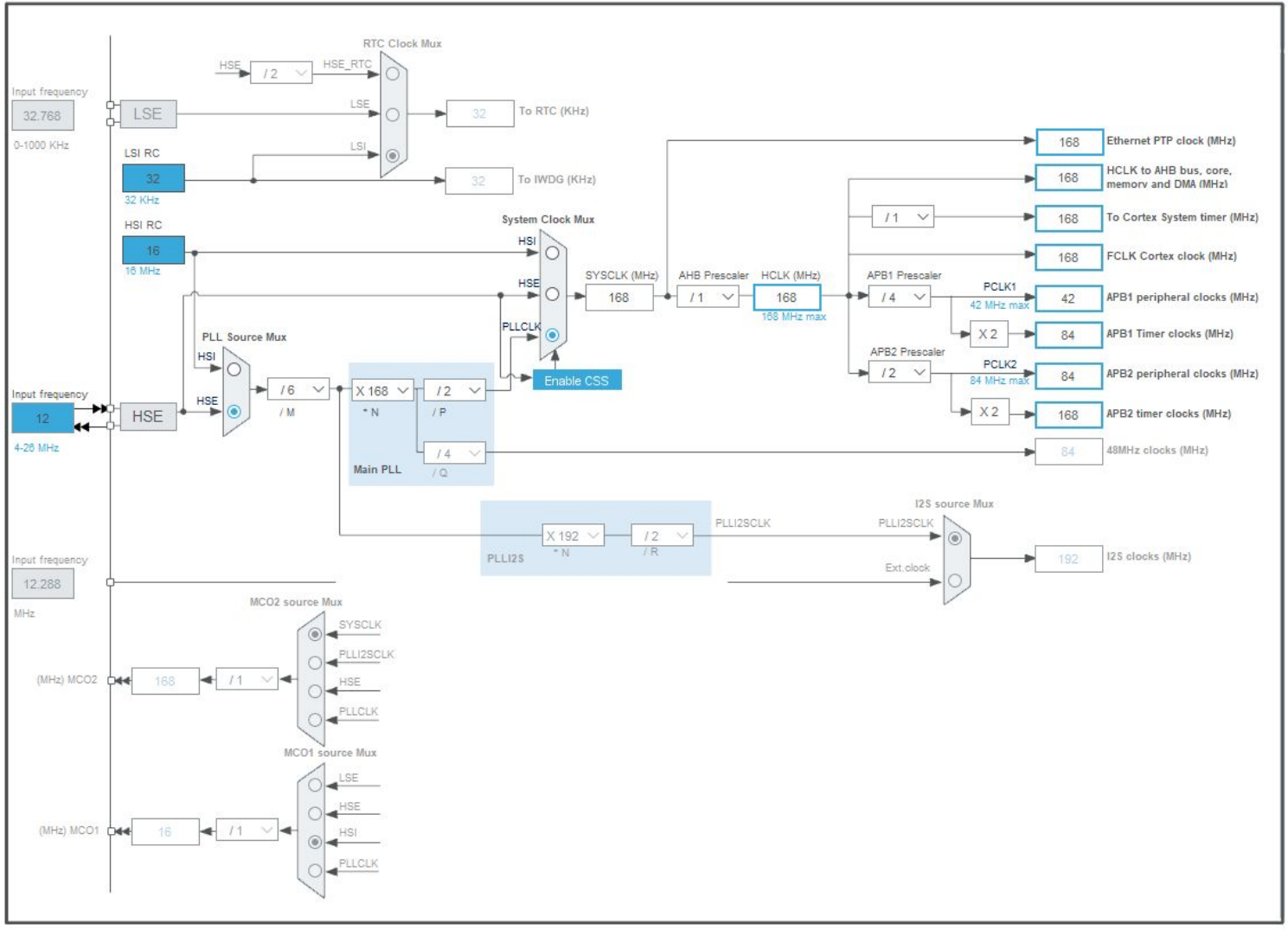
/P = /2

Select PLLCLK

APB1 Prescaler = /4

APB2 Prescaler = /2

Check the rest with the next slide


RTC Clock Mux
HSE
/ 2
HSE_RTC
LSE
LSI
Input frequency
32.768
0-1000 KHz
LSE
32
To RTC (KHz)
LSI RC
32
32 KHz
32
To IWDG (KHz)
HSI RC
16
16 MHz
System Clock Mux
HSI
HSE
PLLCLK
SYSCLK (MHz)
168
AHB Prescaler
/ 1
HCLK (MHz)
168
168 MHz max
Enable CSS
PLL Source Mux
HSI
HSE
Input frequency
12
4-26 MHz
HSE
/ 6
/ M
X 168
* N
/ 2
/ P
/ 4
/ Q
Main PLL
APB1 Prescaler
/ 4
PCLK1
42 MHz max
APB2 Prescaler
/ 2
PCLK2
84 MHz max
/ 1
X 2
X 2
168
Ethernet PTP clock (MHz)
168
HCLK to AHB bus, core, memory and DMA (MHz)
168
To Cortex System timer (MHz)
168
FCLK Cortex clock (MHz)
42
APB1 peripheral clocks (MHz)
84
APB1 Timer clocks (MHz)
84
APB2 peripheral clocks (MHz)
168
APB2 timer clocks (MHz)
84
48MHz clocks (MHz)
PLLI2S
X 192
* N
/ 2
/ R
PLLI2SCLK
I2S source Mux
PLLI2SCLK
Ext.clock
192
I2S clocks (MHz)
Input frequency
12.288
MHz
MCO2 source Mux
SYSCLK
PLLI2SCLK
HSE
PLLCLK
(MHz) MCO2
168
/ 1
MCO1 source Mux
LSE
HSE
HSI
PLLCLK
(MHz) MCO1
16
/ 1

Step 4:

Set debug mode

Pinout & Configuration

-> System Core

-> Sys

Select

-> Debug

-> Serial Wire

LED Pin


VCC_3V3_S
AD1
AD2
AD3
AD4
AD5
AD6
AD7
AD8
C3
C4
C5
C6
C7
C8
C9
C10
0201 50V47pF
R6
1K
1%
0201
[10] USART6_RX(PG9)
[6] LASER(PG13)
[10] USART6_TX(PG14)
[8] POWER1_CTRL(PH2)
[8] POWER2_CTRL(PH3)
[8] POWER3_CTRL(PH4)
OSC_IN
OSC_OUT
U1C
N7
M7
L15
K15
K14
K13
J15
J14
H14
C10
B10
B9
B8
A8
A7
B7
G1
H1
F4
G4
H4
PG0/FSMC_A10/EVENTOUT
PG1/FSMC_A11/EVENTOUT
PG2/FSMC_A12/EVENTOUT
PG3/FSMC_A13/EVENTOUT
PG4/FSMC_A14/EVENTOUT
PG5/FSMC_A15/EVENTOUT
PG6/FSMC_INT2/EVENTOUT
PG7/FSMC_INT3/USART6_CK/EVENTOUT
PG8/USART6_RTS/ETH_PPS_OUT/EVENTO
PG9/USART6_RX/FSMC_NE2/FSMC_NCE3/E
PG10/FSMC_NCE4_1/FSMC_NE3/EVENTOU
PG11/FSMC_NCE4_2/ETH_MII_TX_EN/ETH_
PG12/FSMC_NE4/USART6_RTS/EVENTOUT
PG13/FSMC_A24/USART6_CTS/ETH_MII_TX
PG14/FSMC_A25/USART6_TX/ETH_MII_TXD
PG15/USART6_CTS/DCMI_D13/EVENTOUT
PH0-OSC_IN(PH0)/EVENTOUT/OSC_IN
PH1-OSC_OUT(PH1)/EVENTOUT/OSC_OUT
PH2/ETH_MII_CRS/EVENTOUT
PH3/ETH_MII_COL/EVENTOUT
PH4/I2C2_SCL/OTG_HS_ULPI_NXT/EVENTO

Step 5:

Enable GPIO for LED

Go to “Pinout &Configuration”

At bottom right, in the search box, type “PG1”

Find the circle named “PG1”

Click on it, select “GPIO_Output”

On the left pane, go to “System Core” -> “GPIO”

Give the pin a meaningful name under “User Label” other than “PG1”

Step 6:

Project config

Go to “Project Manager”

Enter project name under “Project Name”

Select a location to store project. STM32CubeMX project is saved as a file with “ioc” extension

Select “MDK/ARM” under “Toolchain /IDE”

Select “V5.27” or “V5” under “Min Version”

Step 7:

Code Generator config

Click “Code Generator” on the left pane

Select “Copy only the necessary library files”

Check “Generate peripheral initialization as a pair of ‘.c/.h’ files per peripheral

Save the STM32CubeMX project

Click on “GENERATE CODE” on top right

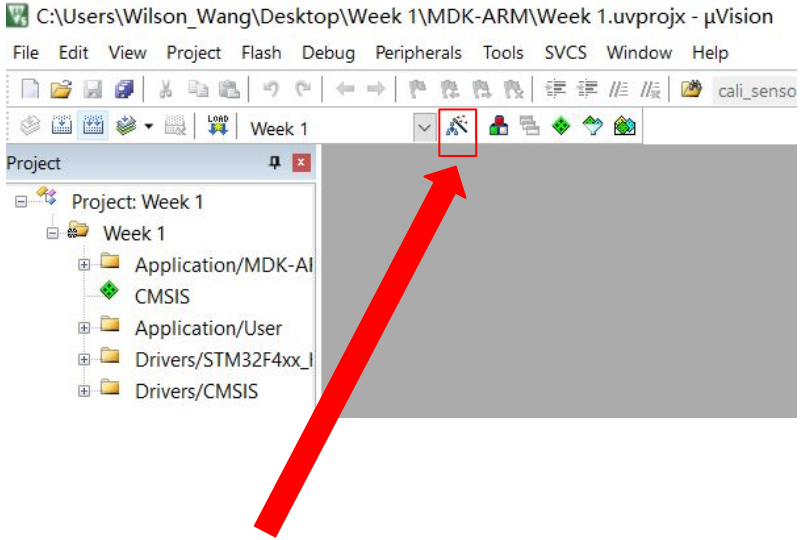
Step 8:

Keil uvision

In the same folder that stores the CubeMX project

Open “MDK-ARM” directory

Open the file with “uvprojx” extension using Keil uvision

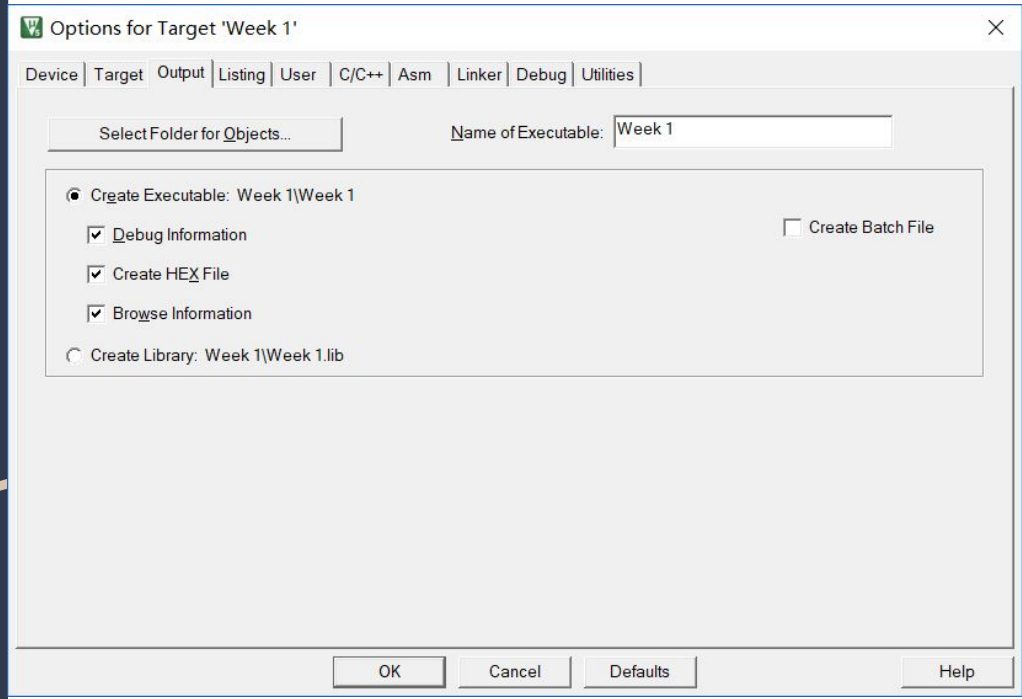


Click to open “Options for Target …”

Step 9/1:

Config Keil

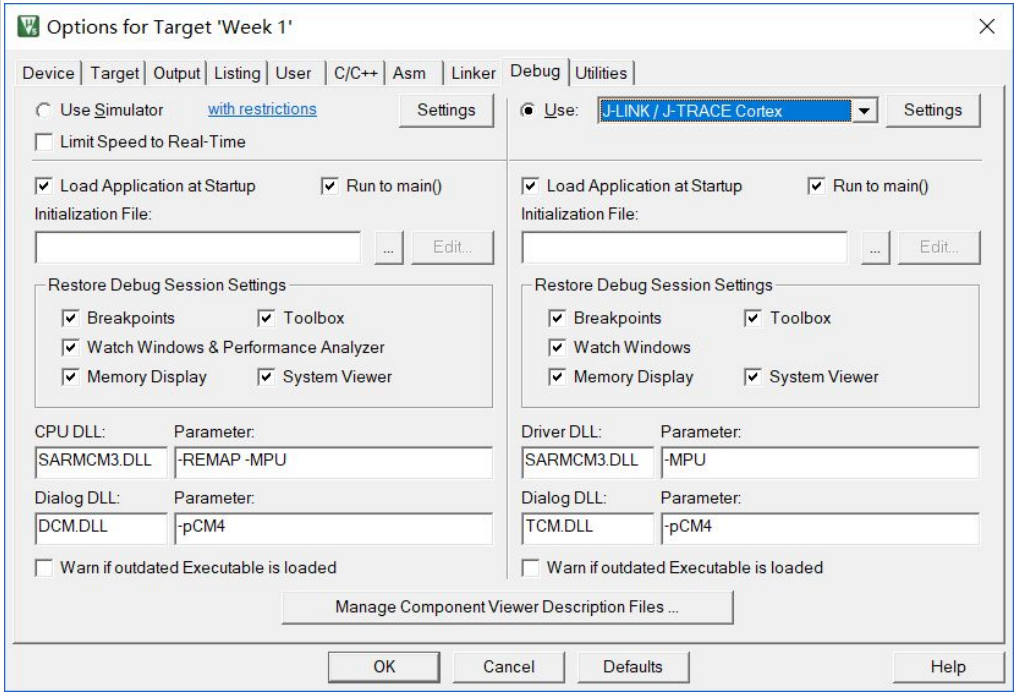
Click on “Output”



Step 9/2:

Config Keil

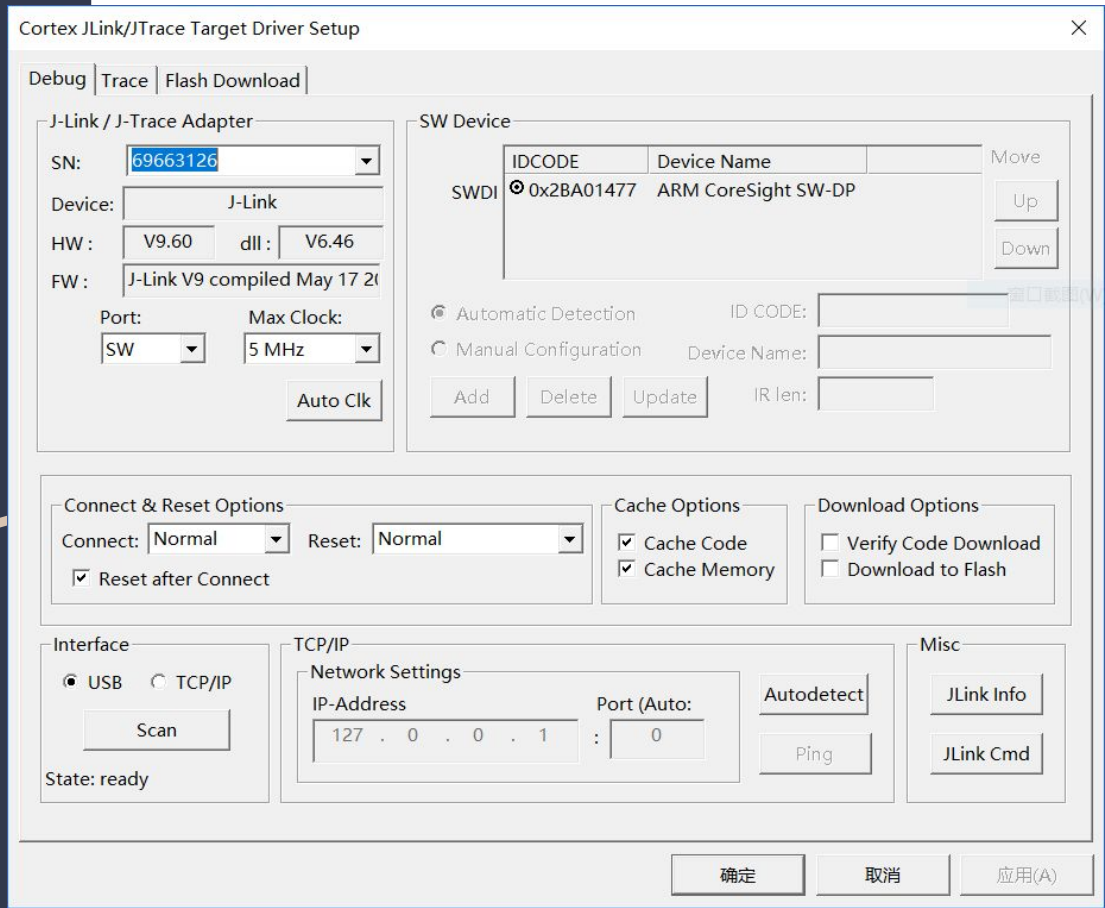
Click on “Debug”



Step 9/3:

Config Keil

Click on Debug -> Settings

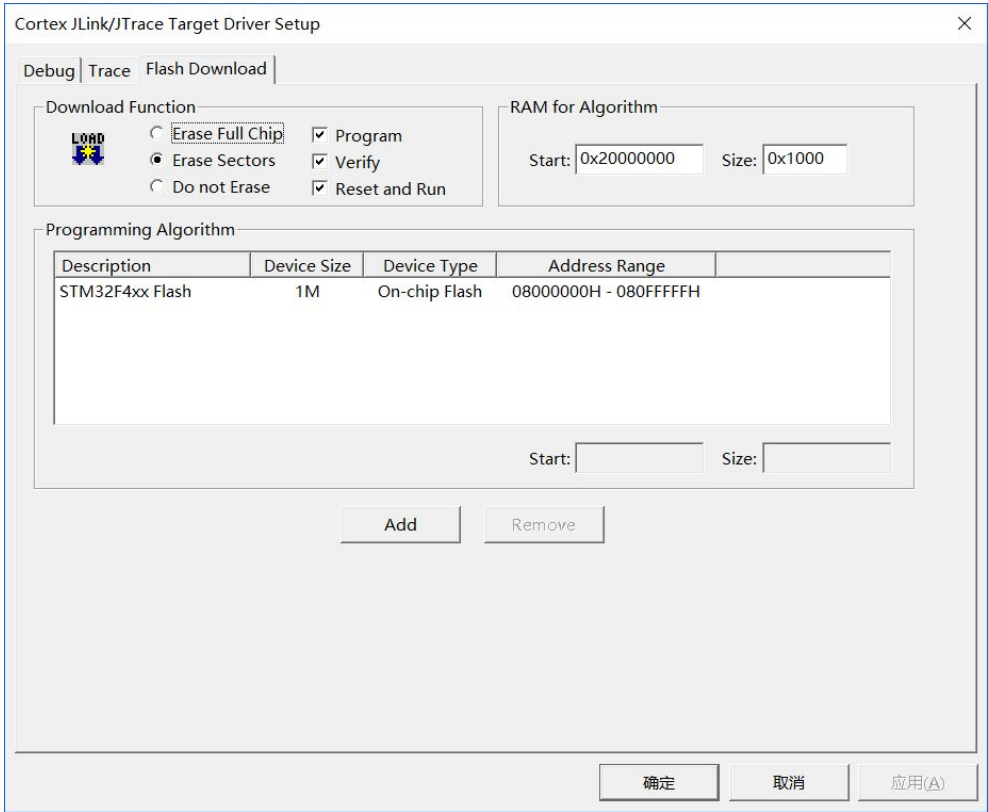


Step 9/4:

Config Keil

Click on Debug -> Settings -> Flask Download

Tick Reset and Run



Step 10:

Coding in Keil

In the file tree view, open “Application/User” -> “main.c”

Go to line 101, as you can see, the line is inside a “while(1)” loop

Enter the following code

At line 91:

```
HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, GPIO_PIN_SET);
```

At line 101:

```
HAL_GPIO_TogglePin(LED1_GPIO_Port, LED1_Pin);

HAL_Delay(1000);
```

After compiling and downloading, you should see one LED lighting up and blinking