

EasyMxPRO

Contents

MikroElektronika EasyMx PRO v7

- Documentation
- Features
- Board defaults checklist
- Hardware considerations
- How to create a new CCS project
- Code samples

MikroElektronika EasyMx PRO v7

Documentation

- Manufacturer website (<https://web.archive.org/web/20210114162445/http://www.mikroe.com/easymx-pro/stellaris/>)
- Board manual (https://web.archive.org/web/20210114162445/http://www.mikroe.com/downloads/get/1812/easymx_pro_v7_stellaris_manual_v102.pdf)
- Board schematic (https://web.archive.org/web/20210114162445/http://www.mikroe.com/downloads/get/1813/easymx_pro_v7_stellaris_sch_v102.pdf)
- MCU socket schematic (https://web.archive.org/web/20210114162445/http://www.mikroe.com/downloads/get/1918/mcu_card_lm4f232h5qd_manual.pdf)
- Code Composer Studio IDE v5 (<https://web.archive.org/web/20210114162445/http://www.ti.com/tool/ccstudio>)
- TivaWare™ for C Series (<https://web.archive.org/web/20210114162445/http://www.ti.com/tool/sw-tm4c>)
- Tiva™ TM4C123GH6PGE Microcontroller datasheet (<https://web.archive.org/web/20210114162445/http://www.ti.com/lit/ds/symlink/tm4c123gh6pge.pdf>) (identical to LM4F232H5QD)

Features

EasyMx PRO is a beautiful and solid development board with extensive variety of functions:

- On-board ICD1 emulator
- Buzzer
- Analog input
- Audio codec VS1053B (<https://web.archive.org/web/20210114162445/http://www.vlsi.fi/fileadmin/datasheets/vs1053.pdf>)
- Access to user read or write of most GPIO ports through pin headers, pull up/down switches, buttons and LEDs
- Digital navigation joystick (5-button)
- CAN interface TI SN65HVD230 (<https://web.archive.org/web/20210114162445/http://www.ti.com/product/sn65hvd230>)
- SD card slot
- USB ports (host and device)
- 2x USB serial interfaces
- Color TFT LCD (320x240) with touch controller
- M25P80 (<https://web.archive.org/web/20210114162445/http://www.micron.com/parts/nor-flash/serial-nor-flash/m25p80-vmc6g>) External Flash
- 24AA01 (<https://web.archive.org/web/20210114162445/http://ww1.microchip.com/downloads/en/devicedoc/21711c.pdf>) External EEPROM
- DS1820 (<https://web.archive.org/web/20210114162445/http://www.maximintegrated.com/datasheet/index.mvp/id/3021>) digital temperature sensor
- TI LM35 (<https://web.archive.org/web/20210114162445/http://www.ti.com/lit/ds/symlink/lm35.pdf>) analog temperature sensor
- 2x mikroBUS connectors (<http://www.mikroe.com/click/>)
- TI Tiva TM4C123GH6PGE (<https://web.archive.org/web/20210114162445/http://www.ti.com/product/tm4c123gh6pge>) (former Stellaris LM4F232H5QD)
 - 80 [MHz] 100 DMIPS ARM Cortex-M4F with FPU
 - 256 [kB] Flash
 - 32 [kB] SRAM
 - ROM loaded with TivaWare™ for C Series software
 - 2 [kB] EEPROM
- MCU peripherals:
 - Analog comparators
 - 2x 1 [MSPS] 12 bit ADC with up to 12 channels
 - Temperature sensor
 - 8x UARTs
 - 4x SSI/SPI
 - Host/Device/OTG Full Speed USB
 - 2x CAN A/B controllers
 - 6x I2C



EasyMx PRO v7 board overview

- 2x Quadrature encoders (QEI)
- 16x PWM Outputs
- 12x Timers
- 2x Watchdogs
- 32x channel DMA
- Hibernation system
- Real Time Clock
- 9x GPIO ports
- ARM SWD JTAG

Board defaults checklist

All demos will assume that user had reset the tool hardware to its default state. In case of problems please ensure that all following conditions are met:

1. MCU card
 - Card is properly aligned with markings on the board (bump located on the top, no pinheads visible under card)
2. MikroProg
 - The USB cable is connected to PC and board
 - All jumpers are connected to the left (TCK, TMS, TDI, TDO)
3. Power Supply
 - Jumper J1 selects USB
 - Switch set to ON (Green ON LED should be shining now)
4. Ethernet (The socket is present but there is no Ethernet functionality)
 - Both jumpers are disconnected (LEDA, LEDB)
5. Analog input
 - Jumper is disconnected
6. Switches SW10 - SW15
 - All set to OFF (left position)
7. Button press level
 - Both jumpers disconnected
 - All pull up/down switches set to not connected (middle position)
8. PORTA+PORTJ
 - All pull up/down switches set to not connected (middle position)
9. Other
 - No MMC card in socket
 - No click cards connected
 - No wires, cables or additional jumpers connected to board

Hardware considerations

Please note that following pins are special:

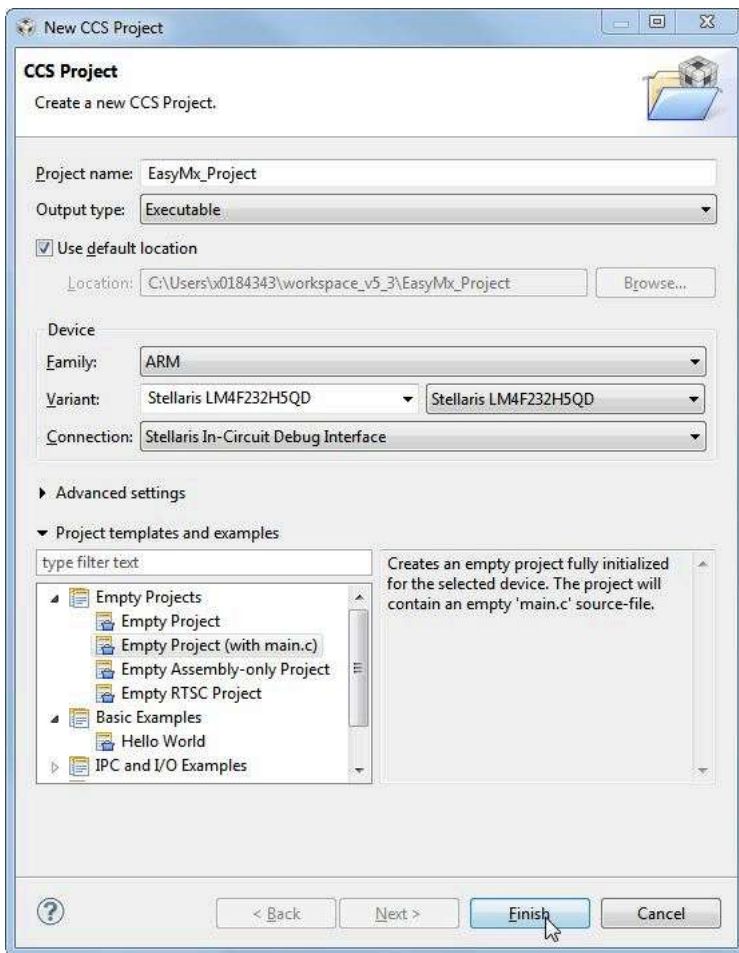
- PORTB[6,7] - not present in MCU
 - Board PB7 is connected to MCU PK7
 - Board PB6 is connected to MCU PK6
- PORTC[0:3] - not connected by default but user can connect those signals to MCU using jumpers in mikroProg section (by default MCU is connected to JTAG through those signals)
- PORTD[0:3]
 - Board PD0 is connected to MCU PN0
 - Board PD1 is connected to MCU PN1
 - Board PD2 is connected to MCU PK4
 - Board PD3 is connected to MCU PK5
- Ethernet socket and magnetics are not connected as MCU does not have Ethernet MAC or PHY

How to create a new CCS project

Written for Code Composer Studio 5.3

New project can be created from scratch or based on existing one. To create a new empty project please follow instruction below. If you want to base your project on some other one select Project → Import existing CCS Project

1. Select File → New → CCS Project
2. Type Project name
3. Select device Variant: Stellaris LM4F232H5QD
4. Select device Connection: Stellaris In-Circuit Debug Interface
5. From Project templates and examples select Empty Project (with main.c)
6. Click Finish



Next step is to add paths for header files and include TivaWare libraries. Please install required software first. In this case Tiva Ware is installed under C:\TI\TivaWare\. To allow easy migration environment variables are used. Changing single variable is much less troublesome than changing path of every external file in project.

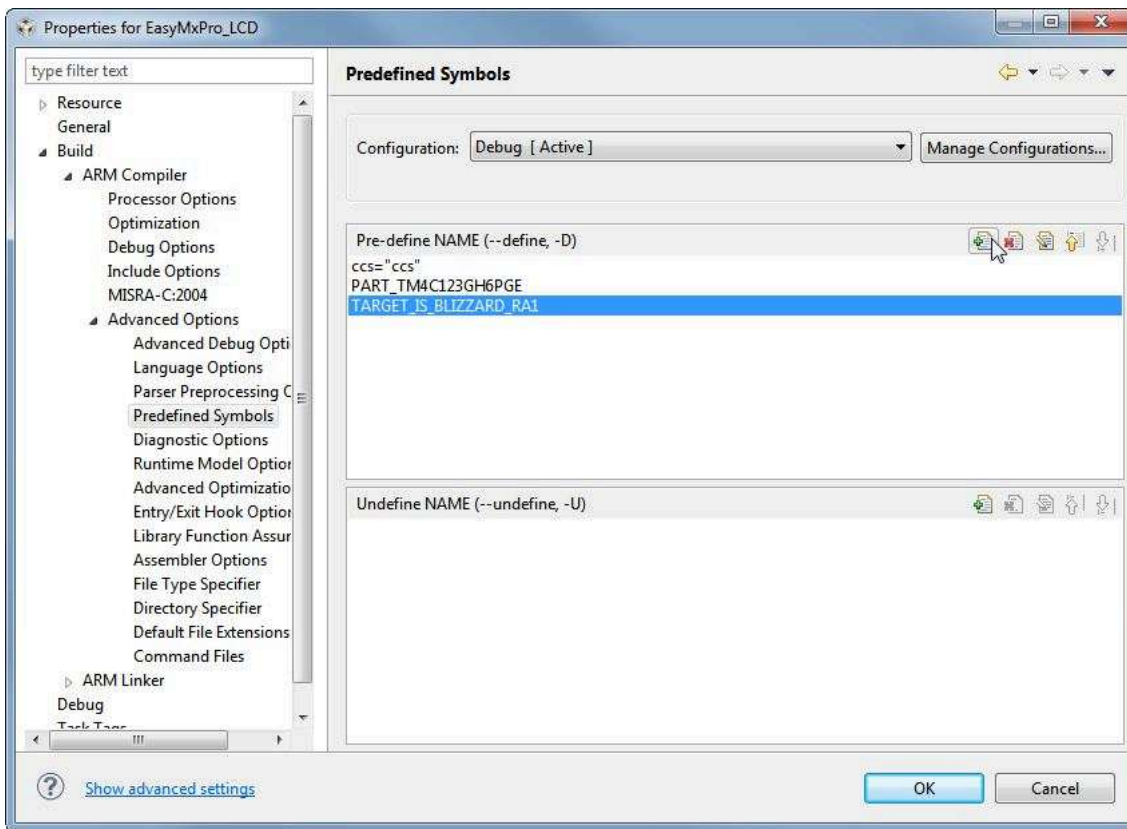
1. Select **Project** → **Properties**
2. Go to tab: **Build** → **Environment** and click button **Add...**
3. Type **Name**: TIVA_WARE and **Value**: C:\TI\TivaWare

When environment variable pointing to TivaWare is defined we can add DriverLib. Compiled code library for handling microcontroller peripherals.

1. Go to tab: **Build** → **ARM Linker** → **File Search Path** find section **Include library file or command file as input**
2. Click add button (with green +) and type \${TIVA_WARE}\driverlib\ccs\Debug\driverlib.lib

Header files are designed to work with different types of devices and compilers. User need to define what compiler and part is using.

1. Go to tab: **Build** → **ARM Compiler** → **Advanced options** → **Predefined Symbols** find section **Pre-define NAME**
2. Click add button (with green +) and type ccs="ccs"
3. Click add button again and type PART_TM4C123GH6PGE
4. Click add button for the third time and type TARGET_IS_BLIZZARD_RA1



Finally we are adding header files location to the project.

1. Go to tab: **Build** → **ARM Compiler** → **Include options** find section **Add dir to #include search path**
2. Click add button (with green +) and type `${TIVA_WARE}`

Now project properties dialog can be closed by clicking **OK**

User can validate if all paths are set correctly by building following project:

```
#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_gpio.h"
#include "inc/hw_memmap.h"
#include "driverlib/sysctl.h"
#include "driverlib/gpio.h"
#include "driverlib/rom.h"

#define GPIO_PINS_ALL GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7

int main(void)
{
    uint8_t a = 0;

    //
    // Set the clocking to run directly from the crystal.
    //
    ROM_SysCtlClockSet( SYSCTL_SYSDIV_20 | SYSCTL_USE_PLL | SYSCTL_OSC_MAIN | SYSCTL_XTAL_8MHZ);


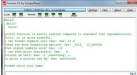


    //
    // Enable GPIOA
    //
    SysCtlPeripheralEnable( SYSCTL_PERIPH_GPIOA);

    //
    // Set all GPIOA pins as outputs
    //
    GPIOPinTypeGPIOOutput(GPIO_PORTA_BASE, GPIO_PINS_ALL);

    //
    // Show some patterns on LEDs
    //
    for (;;)
    {
        //
        // Write data to port
        //
        GPIOPinWrite( GPIO_PORTA_BASE, 0xFF, a++);

        //
        // Delay for a while so changes can be visible
        //
        SysCtlDelay(SysCtlClockGet() / 2);
    }
}
```

Code samples

LEDs	
	<p>This simple demo shows different patterns on the on-board led array.</p> <p>In order to use the demo set board to default state and change SW15 to all-on.</p> <p>File: EasyMx LED.zip</p>
UART	
	<p>This demo shows how to communicate between board and PC through UART interface.</p> <p>In order to use the demo set board to default state, change SW10[1:2] to ON and connect board with PC through USB UART A.</p> <p>One can use different software on PC (examples: Putty, Termite, RealTerm, minicom). Sended string should end with newline character.</p> <p>File: EasyMx UART.zip</p>
LCD	
	<p>This demo uses on-board LCD display. It is based on TI glib. In order to use the demo set board to default state, change SW11 and SW12[2:7] to ON.</p> <p>File: EasyMx LCD.zip</p>
FreeRTOS	
	<p>Demo based on freertos demo from EK-TM4C123GXL Firmware Package. Visit http://www.freertos.org/ for more information on FreeRTOS. In order to use the demo set board to default state, change SW15[8] to [ON], SW10[1:2] to [ON] and connect PC through USB cable to USB UART A.</p> <p>File: EasyMx freertos.zip</p>
USB Host: BMP Slideshow	
	<p>Demo showing Tiva USB Host capabilities. On-board LCD shows bitmap files from connected USB memory.</p> <p>In order to use the demo:</p> <ul style="list-style-type: none"> Set board to default state. Set SW10[7:8] to [ON] Set SW10[1:2] to [ON] for UART Set SW11 to [ON] Set SW12[2:7] to [ON] Connect ANALOG INPUT to PE7 Connect FAT-formatted USB pendrive to USB HOST port Only 256-colour or 24bit BMP images are accepted <p>File: EasyMx Slideshow.zip</p>
USB Device: HID Mouse	
	<p>Sample application utilising on-board touchscreen. Device connected to the PC acts as HID mouse.</p> <p>File: EasyMx Device Mouse.zip</p>

Keystone=

{{
1. switchcategory:MultiCore=

- For technical support on MultiCore devices, please post your questions in the [C6000 MultiCore Forum](#)
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the [BIOS Forum](#)

Please post only comments related to the article **EasyMxPRO** here.

- For technical support on MultiCore devices, please post your questions in the [C6000 MultiCore Forum](#)
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the [BIOS Forum](#)

Please post only comments related to the article **EasyMxPRO** here.

C2000=For technical support on the C2000 please post your questions on [The C2000 Forum](#). Please post only comments about the article **EasyMxPRO** here.

DaVinci=For technical support on the DaVinci please post your questions on [The DaVinci Forum](#). Please post only comments about the article **EasyMxPRO** here.

MSP430=For technical support on MSP430 please post your questions on [The MSP430 Forum](#). Please post only comments about the article **EasyMxPRO** here.

OMAP35x=For technical support on OMAP please post your questions on [The OMAP Forum](#). Please post only comments about the article **EasyMxPRO** here.

OMAPL1=For technical support on OMAP please post your questions on [The OMAP Forum](#). Please post only comments about the article **EasyMxPRO** here.

MAVRK=For technical support on MAVRK please post your questions on [The MAVRK Toolbox Forum](#). Please post only comments about the article **EasyMxPRO** here.

For technical please post y questions at <http://e2e.ti.c> Please post c comments at article **EasyMxPRO** here.}}

Links



[Amplifiers & Linear](#)

[Audio](#)

[Broadband RF/IF & Digital Radio](#)

[DLP & MEMS](#)

[High-Reliability](#)

[Interface](#)

[Processors](#)

- [ARM Processors](#)

- [Digital Signal Processors \(DSP\)](#)

[Switches & Multiplexers](#)

[Temperature Sensors & Control ICs](#)

[Wireless Connectivity](#)

[Clocks & Timers](#)
[Data Converters](#)

[Logic](#)
[Power Management](#)

- [Microcontrollers \(MCU\)](#)
- [OMAP Applications Processors](#)

Retrieved from "<https://processors.wiki.ti.com/index.php?title=EasyMxPRO&oldid=158142>"

This page was last edited on 4 August 2013, at 10:58.

Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.