

Instructions for porting CAN commands into MMBasic 3.2c for use on DuinoMite Mega

Please respect all license agreements and do not distribute CAN.c/CAN.h, MMBasic Source or hex files generated by this process. What you produce from this process is for your personal, non-commercial use ONLY.

These instructions come with no warranty, and no support. PROCEED AT YOUR OWN RISK.

Step One: Get the Required Files

- 1) Request MMBasic Source code from Geoff Graham: <http://mmbasic.com/source.html>
- 2) Get a copy of CAN.c & CAN.h from here: <https://github.com/OLIMEX/DuinoMite/tree/master/SOFTWARE/DMBasic/src/Source/DuinoMite>
There are a number of ways to get the files:
 - a) Install a Git client (such as <http://windows.github.com/>) and then use that to clone the DuinoMite repository
 - b) Choose the “download as zip file” button on this page: <https://github.com/OLIMEX/DuinoMite>
 - c) View each file and copy and paste the source into a text editor and then save them as CAN.c and CAN.h
<https://github.com/OLIMEX/DuinoMite/blob/master/SOFTWARE/DMBasic/src/Source/DuinoMite/CAN.c>
<https://github.com/OLIMEX/DuinoMite/blob/master/SOFTWARE/DMBasic/src/Source/DuinoMite/CAN.h>
- 3) Download & Install MPLAB IDE v8.80 from here: http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en023073
(again, make sure you get MPLAB IDE v8.80)

Step Two: Add the files to the MMBasic Project

- 1) Unzip the MMBasic 3.2c source code and then using File Explorer navigate to: C:\MMBasic\Maximite\MPLAB (or wherever you put the files on your PC). Double click the DUINOMITE.mcp project. If all goes well then this should open up MPLAB IDE v8.80 with the DUINOMITE project loaded. If it doesn't then make sure MPLAB IDE is installed correctly.
- 2) Copy CAN.c & CAN.h over into the MMBasic source code. You can put them wherever you like. I created a new folder called CAN and put them in there: C:\MMBasic\Maximite\CAN
- 3) Update the DUINOMITE.mcp project to add in the CAN.c & CAN.h files. Easiest way to do this is to right click on the “Source Files” folder and choose Add Files, and then right click on “Header Files”. Or you can choose Project-> Add Files to Project from the menu. Remember to add both the CAN.c & CAN.h

Step Three: Edit the files

- 1) In CAN.h change the two occurrences of
`#define INCLUDE_FUNCTION_DEFINES`
To
`#if !defined(INCLUDE_COMMAND_TABLE) && !defined(INCLUDE_TOKEN_TABLE)`
- 2) Edit C:\MMBasic\Maximite\MMSource\Hardware_Includes.h to include the CAN.h file. The include path needs to point to wherever you put the files. For me it looks like this:
`#ifdef DUINOMITE`
`// JDH`

```
#include "../CAN/can.h"  
#endif
```

- 3) Edit C:\MMBasic\Maximite\MMSource\Main.c to make a call to CanInit(). Find the function call INTEnableSystemMultiVectoredInt(); There's only one call like this and it's within the main function. Directly after this command insert CanInit();

```
INTEnableSystemMultiVectoredInt(); // allow vectored interrupts
```

```
#ifndef DUINOMITE  
    CanInit(); // JDH  
#endif
```

- 4) That's all you need to edit. Now choose File->Save All

Step Three: Build the HEX

- 1) In the toolbar at the top of the screen make sure to select the "Release" build. If you see a drop down that says "Debug" click it and select "Release" – you can ignore the pop-up dialog (just click OK).
- 2) Now choose Project -> Build All from the main menu. You should see a message "BUILD SUCCEEDED". If you don't then you need to check the edits you made in Step Two.
- 3) If the build succeeded then you will now find a file called C:\MMBasic\Maximite\Output\DUINOMITE.hex this is the file that you need to flash onto your DuinoMite Mega. Follow the instructions provided by Olimex to flash this software onto your DuinoMite.