



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2016/11.28.13.42-MAN

GUIA PARA OPERAÇÃO DO CUBECOMPUTER

Charles Pereira de Araújo

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34P/3MSH9SL>>

INPE
São José dos Campos
2016

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@inpe.br

COMISSÃO DO CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELECTUAL DO INPE (DE/DIR-544):

Presidente:

Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação (CPG)

Membros:

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

Dr. André de Castro Milone - Coordenação de Ciências Espaciais e Atmosféricas (CEA)

Dra. Carina de Barros Melo - Coordenação de Laboratórios Associados (CTE)

Dr. Evandro Marconi Rocco - Coordenação de Engenharia e Tecnologia Espacial (ETE)

Dr. Hermann Johann Heinrich Kux - Coordenação de Observação da Terra (OBT)

Dr. Marley Cavalcante de Lima Moscati - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Silvia Castro Marcelino - Serviço de Informação e Documentação (SID) **BIBLIOTECA DIGITAL:**

Dr. Gerald Jean Francis Banon

Clayton Martins Pereira - Serviço de Informação e Documentação (SID)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Simone Angélica Del Ducca Barbedo - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Marcelo de Castro Pazos - Serviço de Informação e Documentação (SID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21b/2016/11.28.13.42-MAN

GUIA PARA OPERAÇÃO DO CUBECOMPUTER

Charles Pereira de Araújo

URL do documento original:

<<http://urlib.net/8JMKD3MGP3W34P/3MSH9SL>>

INPE
São José dos Campos
2016



Esta obra foi licenciada sob uma Licença Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada.

This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License.



Ministério da
Ciência, Tecnologia
e Inovação



CONASAT
CONSTELAÇÃO DE NANO SATÉLITES AMBIENTAIS

CONSTELAÇÃO DE NANO SATÉLITES PARA COLETA DE DADOS AMBIENTAIS

RELATÓRIO TÉCNICO

Guia para operação do CubeComputer

CNS-MNL-SY-04-001-V00.00

Julho, 2016

Registro de Propriedade do Documento

Autores	Organização	Data	Assinatura
Charles Pereira de Araújo	INPE/CRN	13/07/2016	

Aprovação	Organização	Data	Assinatura
Manoel Jozeane Mafra de Carvalho	INPE/CRN	__/__/2016	

Histórico de Modificações

Edição	Data	Modificações	Visto

ÍNDICE

Conteúdo

1. RESUMO	ERROR! BOOKMARK NOT DEFINED.
2 - CONTEÚDO	4
3 - CONFIGURANDO O SOFTWARE	4
3.1- INSTALANDO O COMPILADOR.....	4
3.2- INSTALANDO O SIMPLICITY STUDIO.....	5
3.3- ECLIPSE E O WORKSPACE.....	6
4- BUILDING, LOADING E DEBUGGING UMA APLICAÇÃO	7
5- SOURCE APPLICATION OVERVIEW.....	9
5.1- BACKGROUND.C	9
5.2- COMMS.C	9
5.3- TEST.C.....	10
5.4- TEST.C.....	10
6- CRIANDO UM SOFTWARE CUSTOMIZADO.....	11
6.1- RESTRIÇÕES DO HARDWARE.....	11
6.2- PIGGYBACK HEADER.....	12
6.3- MAKEFILE.....	12
7. DOCUMENTAÇÃO ÚTIL	13

LISTA DE ABREVIações

BSP	Board Support Package
ESL	Electronic Systems Laboratory
MCU	Microcontrolador
GPIO	General Purpose Input Output

LISTA DE FIGURAS

Figura 1 : Opção a ser escolhida	4
Figura 2: Interface do Simplicity Studio.....	5
Figura 3 : Pacotes a ser instalados	5
Figura 4 : MCU a ser utilizado	6
Figura 5 : Area de desenvolvimento do Eclipse	7
Figura 6 : MCU em modo Debug OUT	7
Figura 7 : Conexão dos equipamentos	8
Figura 8 : Configuração do GDB Server	8

1. Escopo

Este presente relatório tem como objetivo apresentar um resumo breve de como funciona o *Building*, *Loading* e o *Debugging* de uma aplicação usando o CubeComputer, desde a instalação dos software ate o manuseio com o equipamento. Esse relatório foi baseado no relatório General Purpose Onboard Computer.

2 - Conteúdo

O CubeComputer, apresentado nesse relatório, é fornecido com o software para programar e desenvolver o código para o CubeComputer. Bibliotecas e código de teste é fornecido em um espaço de trabalho para o usuário para começar com CubeComputer. Segue a descrição do acervo contido no CD:

Pasta	Conteúdo
\CubeComputerBSP\documentation	Contém uma coleção de documentos úteis sobre o CubeComputer
\CubeComputerBSP\eclipse	Contém as configurações e os plug-ins do ambiente de integrado para desenvolvimento do software
\CubeComputerBSP\installations	Contém todos os executáveis necessários para a instalação dos programas
\CubeComputerBSP\workspace\	Diretório para usar o eclipse
.\workspace\metadata\	Contém todas as configurações para uso do Eclipse
.\workspace\libraries\	Contém todas as bibliotecas externas para a aplicação do CubeComputer

3 - Configurando o Software

Copie todo o conteúdo “CubeComputerBSP” para o seu ambiente de trabalho.

1.1 3.1- Instalando o compilador

O compilador usado foi o “*Sourcery CodeBench Lite for ARM/EABI*”. Para instalar o compilador, execute-o no seguinte diretório : “CubeComputerBSP\Intallations\arm-2013.05-23-arm-none-eabi.exe”

Quando você for questionado, no ato da instalação, sobre a adição e/ou modificação da variável de trabalho, deve-se escolher a opção (Modify PATH for all users)como mostra a figura a seguir :



Figura 1 : Opção a ser escolhida

3.2- Instalando o Simplicity Studio

O Simplicity Studio centraliza todas as aplicações, programas de exemplo e documentação para o MCU usado no CubeComputer. Execute o aplicativo “Install Simplicity Studio.exe” no diretório "CubeComputerBSP\Installations". O Simplicity Studio também irá instalar o SEGGER JLink GDB Server, que vai ser utilizado para ligar ao MCU através da interface Jlink.

Uma vez instalado, clique em "Adicionar / Remover Pacotes" ícone, baixe e instale os produtos EFM32 no Simplicity Studio.

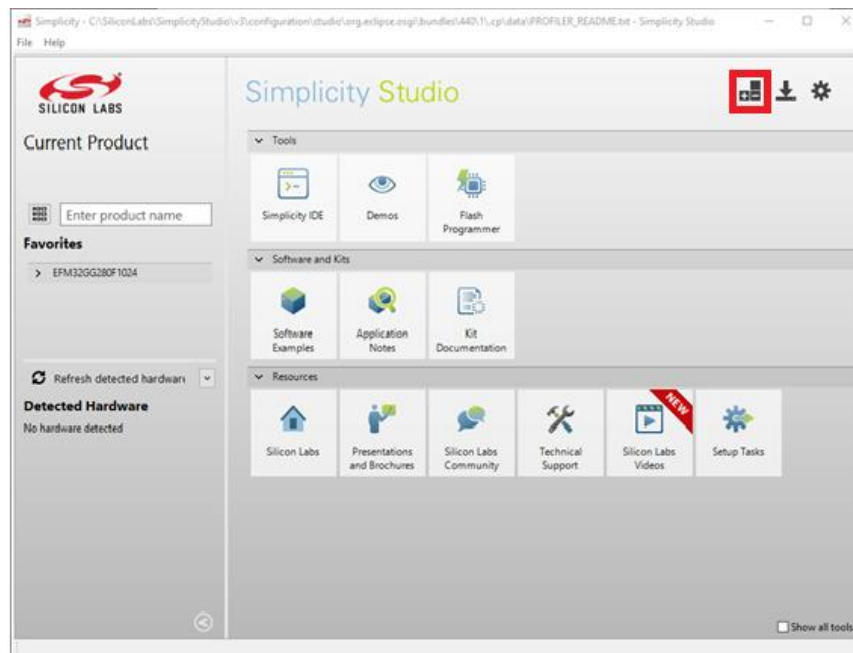


Figura 2: Interface do Simplicity Studio

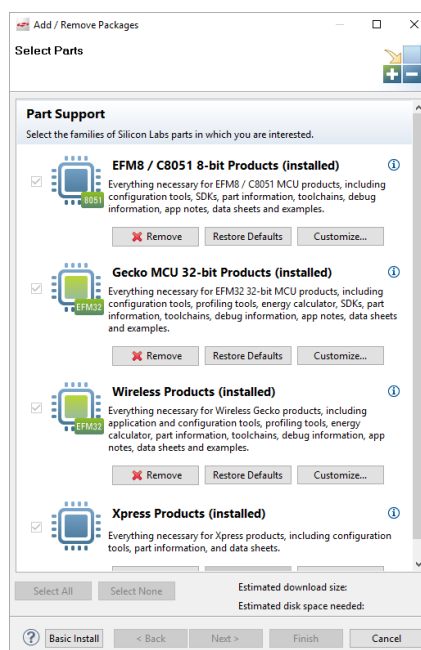


Figura 3 : Pacotes a ser instalados

Especifique o MCU correto (EFM32GG280F1024) no campo "Produto" no Simplicity Studio para ter acesso ao Manual de Referência correta e Folha de Dados. O campo produto é marcada em Figura 4.

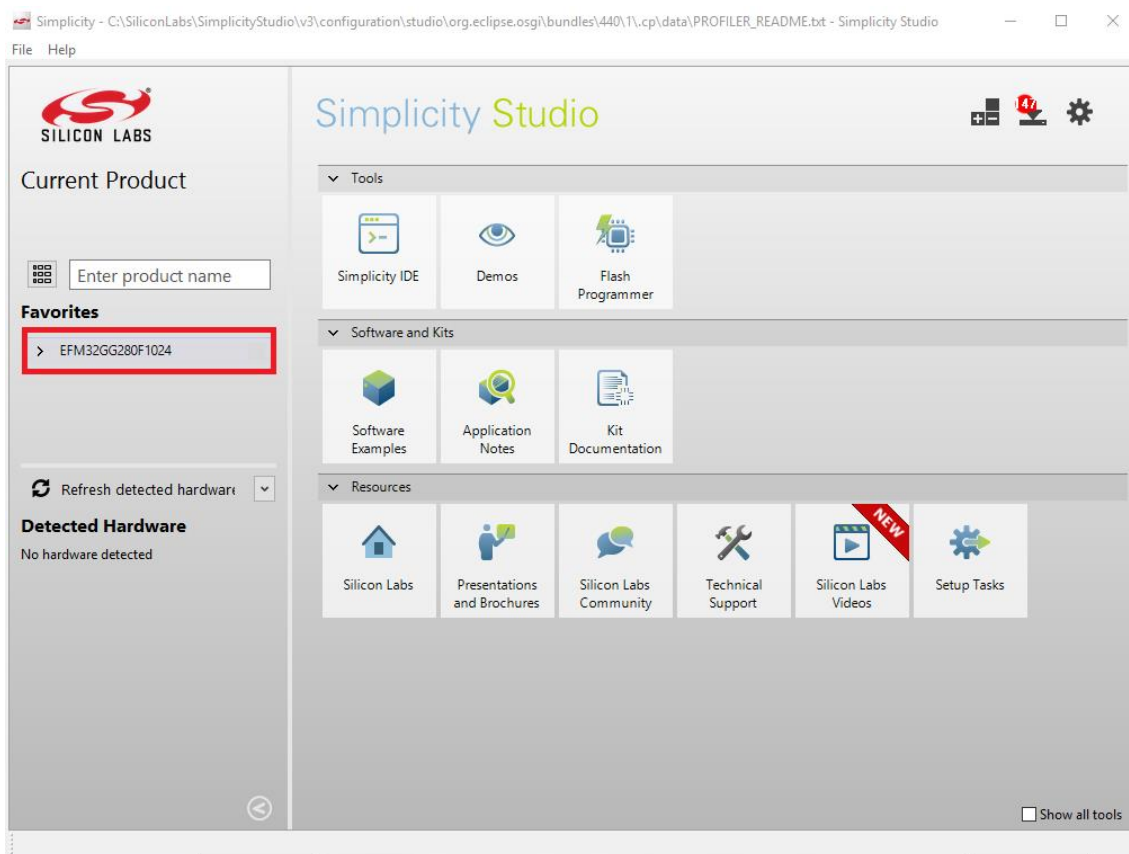


Figura 4 : MCU a ser utilizado

PS: Não atualizar o firmware do MCU.

3.3- Eclipse e o Workspace

Execute "`\\CubeComputerBSP\eclipse\eclipse.exe`" e defina o espaço de trabalho(workspace) do eclipse para "`\\ CubeComputerBSP \ workspace`", como mostrado na Figura 5. Eclipse requer Java Runtime Ambiente e exigirá que você instalá-lo antes de poder utilizar eclipse.

O ambiente eclipse já foi configurado para trabalhar com o compilador e é capaz de programar o CubeComputer. A arquivo em PDF "`AN0023 EFM32 eclipse toolchain`" descreve como eclipse foi instalado e é fornecido no diretório "`Documentation`".

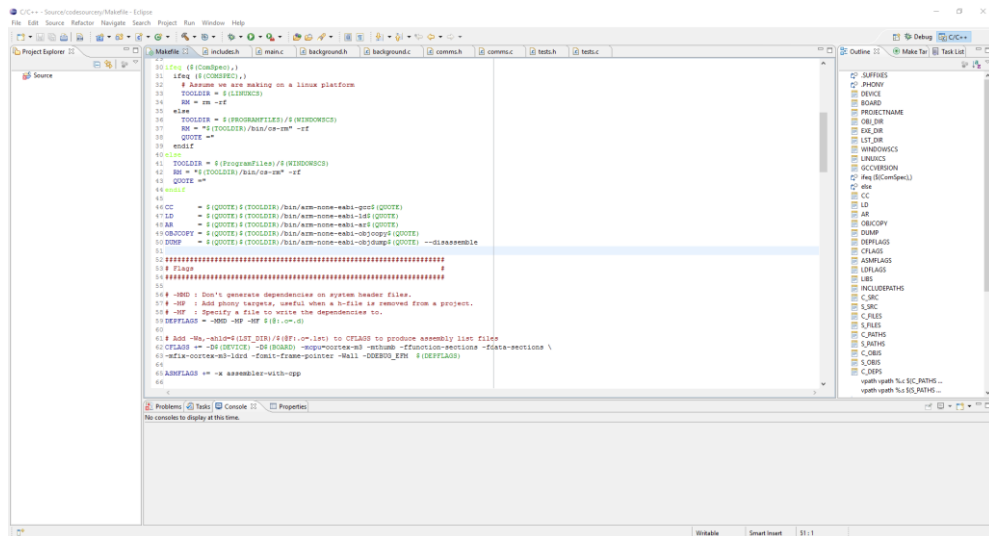


Figura 5 : Área de desenvolvimento do Eclipse

4- *Building, Loading e Debugging* uma aplicação

1. Abra o Eclipse e clique em Build Project que se encontra em:

Project – Build Project

2. Certifique-se que tanto o J-Link debugger está conectado ou o kit Starter é configurado como um debugger. O link a seguir, mostra como conectar o kit start como debugger:

<http://community.silabs.com/t5/32-bit-MCU/Debugging-with-the-EFM32-Starter-Development-Kits/m-p/97969#M3>

3. Abra o Simplicity Studio e clique em *Kit Manager application*
4. Defina o modo do debugger do kit starter, se você estiver usando um kit inicial para programar o CubeComputer. Certifique-se que o starter kit "Debug Mode" está definido para "OUT",. Certifique-se de que o LED "DEBUG OUT" está ligado, no MCU, conforme a figura 6.



Figura 6 : MCU em modo Debug OUT

5. Verifique se o CubeComputer está recebendo 3,3 V de alimentação. Consulte o *datasheet* para obter detalhes sobre as conexões para alimentação e terra.
6. Ligue o cabo UART no P101. Consulte o *datasheet* para obter detalhes sobre os pinos.

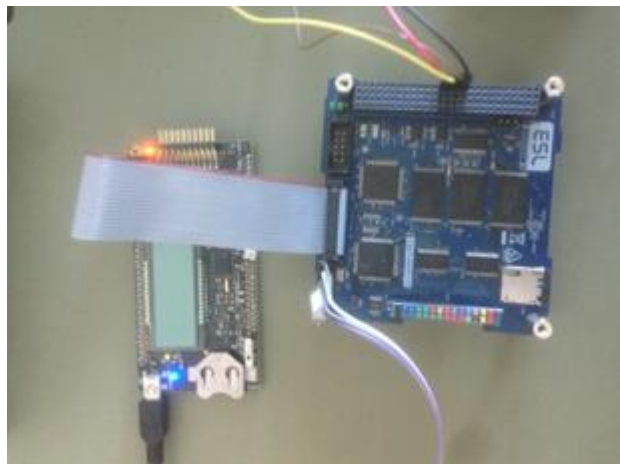


Figura 7 : Conexão dos equipamentos

7. Certifique-se que o GDB Server esta conectado e funcionando, como a figura a seguir :

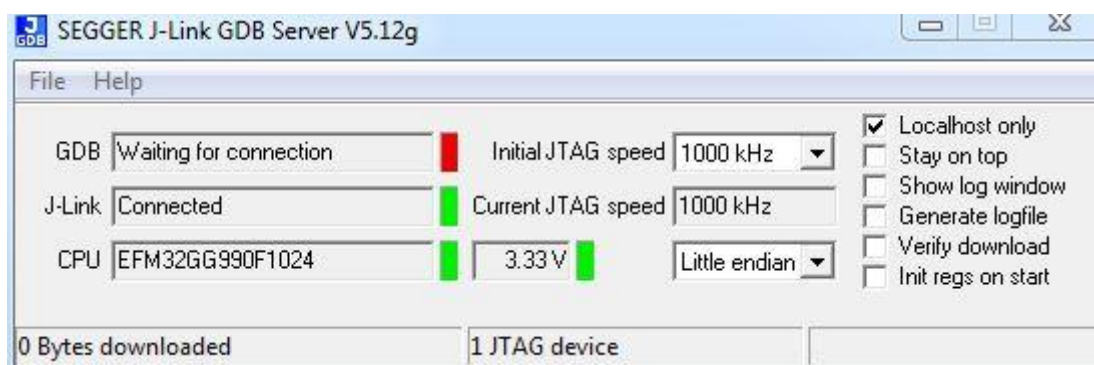


Figura 8 : Configuração do GDB Server

8. Inicie a configuração debugger para o MCU e execute o aplicativo no modo debugger:

Run – Debug History – Source

5- Source Application Overview

O código *main.c* contém um aplicativo de teste simples que mostra o uso da biblioteca de driver para operar os diferentes periféricos e subsistemas de CubeComputer.

O projeto de origem contém um programa simples que pode ser usado como uma base para o desenvolvimento do software. O programa consiste em 3 partes:

5.1- Background.c

Este arquivo contém as rotinas de serviço de interrupção (*IRQs*) para as tarefas de fundo responsáveis pela segurança do subsistema do Cubecomputer, como o circuito anti-latchup, watchdog interno/externo e correção e detecção de erros da SRAM.

5.2- Comms.c

Este arquivo contém os *IRQs* para os periféricos de comunicação, como UART, I2C e CAN. Um exemplo de protocolo unificado (Tabela 1) foi implementado entre os diferentes periféricos Comms. Isso permite que o usuário possa se concentrar apenas na atualização e processar as diferentes mensagens de telecomando e telemetria.

<i>Message Type</i>	<i>ID</i>	<i>Data direction</i>
Telecommand	0x00 – 0x7F	Data Received
Telemetry	0x80 – 0xFF	Data Transmitted

5.3- Test.c

Um resumo de todas as sub-rotinas de teste estão listados abaixo na tabela abaixo:

<i>Subroutine</i>	<i>Descrição</i>
TEST_RTC()	Testa o clock para a saída UART.
TEST_EBI()	Testa a interface de barramento externo através da leitura e escrita para todos os módulos de memória externos e produz os resultados na UART.
TEST_I2C()	Testa a interface I2C, lendo e escrevendo para um módulo externo EEPROM I2C de saída e os valores ao longo do UART. O EEPROM não é fornecido com o CubeComputer.
TEST_ADC()	Testa a interface analógica – digital por amostragem dos canais de tensão, corrente e temperatura e os valores de saída na UART.
TEST_microSD()	Testa a interface para cartão micro SD, abrindo um arquivo de texto e escrevendo um string dentro dele.

A função TEST_I2C () exige que o I2C seja iniciado no modo master. Exige também que o EEPROM externa seja ligado ao CubeComputer. A função também precisa de ser adicionados à lista de comando comms.c a ser chamado pelo usuário através do link UART.

Os seguintes hardwares são recomendados para as funções de teste:

- ST M24C64 I2C externo EEPROM.
- BV4221-V2 USB para conversor de I2C.
- UM232R UART ao conversor USB.

5.4- Test.c

Este é o lugar onde o usuário deve adicionar o código do aplicativo. O loop principal ", while ()" deve ser visto como o ciclo do software de voo que é executado todos os comandos em "x", onde "x" é a quantidade de segundos. Dentro entre malhas de controle, o MCU deve esperar em um estado de baixa energia pronto para processar qualquer telecomando a ser recebido.

6- Criando um software customizado

O CubeComputer é baseada em torno da Energia Micro MCU EFM32GG280F1024. Isto é, portanto, possível desenvolver software personalizado para o CubeComputer usando alguns drivers, notas de aplicação e programas de exemplo disponíveis no Micro Energia (agora Silicon Labs) website. O usuário pode incentivar a usar o projeto fonte como ponto de partida, para minimizar o tempo de instalação, e usar a biblioteca BSP fornecido na medida do possível.

6.1- Restrições do Hardware

Ao criar um aplicativo personalizado é sempre importante levar em consideração as limitações de hardware, configurações essas que serão mostradas na tabela a seguir:

Peripheral	Description
UART0	Assigned to location 0 and is routed to the PiggyBack header P600,P7,P9. Can also be used as GPIO pins.
UART1	Assigned to location 2 and routed to P101.
USART1	Assigned to location 1, SPI mode and routed to CAN controller. (CubeComputer V3B)
USART2	Assigned to location 1, SPI mode and routed to microSD card.
I2C0	Assigned to location 2, routed to H1,P41,P43.
I2C1	Assigned to location 0, can be routed to H1,P1,P23 or P600,P8,P10 or both depending on user selection.
EBI	Assigned to location 0 (AD0-15, CS0-4, ALE, nRE, nWE). Configured to work in multiplexed, 8-bit Data, 24-bit address Mode.
ADC0	CubComputerV3A: channels 0,1,4,5 used, available on PBH CubeComputerV3B: channels 4,5,6,7 used, available on PBH
ACMP0	channel 2 (SRAM1 current)
ACMP1	channel 0 (SRAM2 current)
GPIO	CubeComputerV3A: PD7, power for SRAM1, enable high CubeComputerV3B: PC0, power for SRAM1, enable high
	CubeComputerV3A: PD8, power for SRAM2, enable high CubeComputerV3B: PC1, power for SRAM2, enable high
	PC14, enable for SRAM1, enable low
	PC15, enable for SRAM2, enable low
	PF9, Watchdog enable & toggle
	PE2,3, FPGA control pins
	PB0,1,2, FPGA error pins
	PC11,12,13 CAN control pins

6.2- PiggyBack Header

A PBH (PiggyBack Header) permite ao usuário criar hardware personalizado que pode ser conectado diretamente ao CubeComputer. A pinagem e funcionalidade dos pinos de cabeçalho PBH estão descritos na Tabela a seguir.

PBH Pin	MCU Pin	Signal	Description
1,2	-	V_Bat	Battery supply from main header
3,4	-	5V_MH	5V supply from main header
5,6	-	3V3_MH	3.3V supply from main header
7	PF7	UART_RX	Miscellaneous UART RX
9	PF6	UART_TX	Miscellaneous UART TX
8	PC5	I2C_SCL	Subsystem I ² C clock signal
10	PC4	I2C_SDA	Subsystem I ² C data signal
11	PE5	SPI_CLK	Miscellaneous SPI clock signal
13	PE4	SPI_CS	Miscellaneous SPI chip select
15	PE7	SPI_MOSI	Miscellaneous SPI master out/slave in
17	PE6	SPI_MISO	Miscellaneous SPI master in/slave out
12,14, 16,19	PA8-10, PB11	PWM	PWM output signals
18,20, 21,22, 24	PA11-13, PB12, PE0	GPIO	General purpose input/output
23	PD8	BU_VIN	This pin is used to power the backup power domain. The backup power domain is used by the Backup Real Time Clock.
25-28	PD4-PD7	ADC	12-bit ADC. Connected to the MCU through a voltage follow op-amp circuit. These pins cannot be used as GPIO pins
29,30	-	GND	

6.3- Makefile

Ao criar software personalizado para o CubeComputer, certifique-se de adicionar esses arquivos para o lista de C-arquivos no makefile localizado em "`\" CubeComputer \ workspace \ Source \ CodeSourcery \`". E se os arquivos `.c` e `.h` não estão localizados na pasta de origem do projeto, o seu caminho de inclusão deve também ser adicionado no makefile.

7. Documentação útil

A documentação a seguir é útil na compreensão e funcionamento do CubeComputer e seus subsistemas. Estes documentos podem ser encontrados na subpasta documentação no CubeComputer pacote suporte placa: "\\ CubeComputerBSP \ documentation \".

- MCU EFM32GG Reference Manual MCU EFM32GG280 Datasheet
- AN0023 EFM32 Eclipse Toolchain
- AN0062 EFM32 Programming guide
- AN01 CubeComputer Bootloader
- CubeComputer Data Sheet
- CubeComputer CAN Documentation
- External Watchdog Data Sheet
- CubeComputer System I2C Health Check