PIC4BBB cape

General description

This is free software for embedding of Microchip dsPIC33EP / PIC24EP in BeagleBone Black.
The complex and very bad documented real time units (PRUs) in BBB are used only to program extra "cape" with fast, low power, rich of peripherals and very low cost microcontrollers (real time units) of Microchip. The reprogramming can be executed at any time from the main (Linux, Python) application, only binary code file is used. So the project can be secure, with "on-fly" easy updating via network, external memory, etc.
To use this advanced hardware, the user do not must be embedded Linux expert, no to dig in thousands pages documentation and experimenting which BBB hardware how is used and how can be accessed from the user code (usually it is not possible to access timers, encoder and so on, without complex kernel programming...). So the suggested project make such a simple task as frequency measuring very easy with several rows simple code, in contrary of using long PRU assembler in very not efficient way (software counter).
Some advanced features of the real time unit in the suggested PIC4BBB cape are:
- As you see on the circuit of the cape – dsPIC33EP64MC504 is used.
- fast 16 bit core at frequency up to 70MHz with advanced DSP functions and very low power consumption
- complex up to 6 channel PWM with resolution of about 7ns, proper for advances motor control and power inverters
- fast (above 1Msps) ADC, 10bit or 12bit, with special control, synchronized with advanced PWM or other peripheral
- timers, comparators
- additional UARTs, SPI, I2C, ECAN and others.
All these advanced and specialized peripherals are independent form the BBB, but can be controlled from this main computer.
- The Microchip's controllers are one of the cheapest.
- Has FREE advanced IDE and high-level programming tools (C programming).
- This controllers are very well documented and because of popularity – with good support.
- These PIC types has no internal EEPROM, but BBB has enormous fast memory.
The low level program for PIC can be easily developed and optimized and the high level large code will be easily developed in BBB, so only advantages of the two worlds are used for easy and effective practice design.
- PIC are one of the most secure chips, also fo industrial purposes, so are the BBB. This combination, I believe, is very competitive to specialized industrial controllers, which has of order of many times higher price and usually do not support contemporary aesthetical user interface and user friendly interface, such as BBB Linux with Touch screen.

The BBB works with updated:
bone-debian-7.11-lxde-4gb-armhf-2016-06-15-4gb.img
This version works fine with LCD touchscreen, starts fastly and easily can be installed with

all extra libraries in the integrated 4GB.
The Python 2.7 is used with preinstalled Adafruit libraries and it is added only the free project PyPRUSS (https://bitbucket.org/intelligentagent/pypruss), which do not work with Python3.

This decision is maximal easy to use - no special requirements to the user software, no linker scripts and so on. Every program, for that PIC can be installed and run with BBB.
The PIC controller do not use Operating System, so it starts almost at the moment when is power supplied. So this project can support high security systems, fastly run, until the PIC part supervise the BBB and vica versa.

ATTENTION!: There are no hardware protections if the user set unproperly connected pins between PIC and BBB. Connecting of two pins, switched to Outputs can destroy the BBB or/and PIC.

Packet content

- Layouts.
This is the layout:

mcape-universal-00A0

which is based on free project "beaglebone-universal-io"-cape-universal
(https://github.com/cdsteinkuehler/beaglebone-universal-io).
The last "Exports all pins not used by HDMIN and eMMC (including audio)".
But in
mcape-universal-00A0
are removed these, used by the cape: 4DCAPE-43T (LCD with touchscreen).
So now it us easy to use:
config-pin
tool for pin configuration.

IMPORTANT! To be possible to install this layout, the HDMI must be disabled in uEnv.txt, because the pin P9_25 is used by this cape.

- pymcephex32.py - Intel HEX32 parser for prepare data from compiled Microchip HEX to use in BBB for programming PIC microcontroller
this is command line tool:
py pymcepheh32.py name.hex
which generates name.pbin to use in BBB.
All Config registers must be programmed. Programmer locks the memory at the end.
The resulting *.pbin file is about 10 times smaller than the *.hex. So it is very
proper for on fly programming via Internet and other also because better security.

- stdicsp - project for TI CCS PRU C compiler. This is real time firmware of the dsPIC33EP / PICEP programmer.
From the compiled program, the files Release\data.bin and text.bin are used in the main application (see below).

- stdicsp...py - script for programming PIC with the *.pbin file

- pic4bbb... - project for Microchip MPLAB X with XC16, the *.hex file is in \dist\default\production

Full installation example

1. Install Linux distro - fastest boot time and proper drivers:
bone-debian-7.11-lxde-4gb-armhf-2016-06-15-4gb.img

It has preinstalled Adafruit's libraries for Python2.

2. Install the PyPRUSS:
The packet must be unpacked in any folder, then:

python setup.py install
export LD_LIBRARY_PATH=/usr/local/lib

3. Copy PIC4BBB layout:
mlncape-universalh-00A0.dtbo
into folder:
/lib/firmware

To work, must in /boot/uEnv.txt:

##Disable HDMI (v3.8.x)
cape_disable=capemgr.disable_partno=BB-BONELT-HDMI,BB-BONELT-HDMIN

(may work only with disabling HDMI Audio, some rows down in the file).

4.Now is time for the PIC4BBB - for example pic4bbb_i2c_withExample (the pic4bbb_uart_withExample can be used in the same way):

- in some folder (in BBB or any PC, the script works with any Python version on Linux and on Windows), put the files:

pymcephex32.py
and the compiled
pic4bbb_i2c.production.hex

the last is in XC16 project folder:
\pic4bbb\pic4bbb_i2c_withExample\pic4bbb_i2c\dist\default\production

Then use command:
python pymcephex32.py pic4bbb_i2c.production.hex

If all is ok, in the folder is generated file:
pic4bbb_i2c.production.pbin

This is used in the BBB for on-fly reprogramming of the PIC4BBB cape:

- copy in some BBB folder the files:
 pic4bbb_i2c.production.pbin (this is example, generated on previous step)
 stdicsp_i2c.py (this is the Python programming script)
 data.bin and text.bin from the PRU project in folder:
\pic4bbb\pic4bbb_i2c_withExample\stdicsp_i2c\Release
This is the BBB real time programmer for PIC.
At least - the python example code:
 pic4bbb_i2c.py

Now you are ready to try PIC4BBB real user design in two simple steps:

Reprogram the PIC4BBB with example code:
command:
python stdicsp_i2c.py pic4bbb_i2c.production.pbin

For some seconds, the PIC4BBB is reprogrammed and started, the example program works - LED blinking is observed.

Now try simple user example:

python pic4bbb_i2c.py

This controls LEDs on PIC4BBB cape and prints the communication reply.

So - the suggested PIC4BBB cape is simple, user friendly and proper for fast and reliable machine control systems design.

Lets we know for any your additional requirements, we are ready to suggest to you special control system project, based on this beautiful and compact design.

LICENSE NOTES:

The PIC4BBB cape software is
Free, open-source, fully featured demonstrative software.

THIS SOFTWARE IS PROVIDED,
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PROVIDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY
OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.