This project BCBSP

Adaptation of BP Control Box hardware to Speed Pulse application

This folder contains schematic circuit diagram, relevant datasheet, engineering notes and complete firmware source code.

Firmware build tools :

* Microchip MPLAP X IDE
* Microchip Code Configurator MCC (MPLAB X plug-in)
* Microchip XC8 v1.42 compiler (both PRO- and standard-mode are okay)

The project structure contains source code files generated by MCC. To build the target op-code, MCC is not needed. To change MCU configuration (such as allocating IO pins, adding timer, changing oscillator) MCC can generate new source code. Since MCC configuration file is included in this repo, calling up MCC (from within MPLAB X) will load the current MCU parameters. The generated code will merge automatically into existing code (such as when re-configuring the MCU).

Key MCU features:

* Microchip PIC18F26K22 28-pin DIP
* On-chip oscillator with x4 PLL is used for 64 MHz clock, 16 MIPS
* 5V power supply (needed for 64 MHz)
* 32 k-word Program Memory (64 k-Byte, 10K cycle endurance)
* 3896 bytes Data Memory (RAM)
* 1024 bytes Data EEPROM (100K cycle endurance)
* External I2C EEPROM 24C512 size 512 k-byte (100K cycle endurance)

Key circuit features:

* Power supply endurance: 8V – 36V operating, cutoff protect 36V – 200V+
* ADC input, 36V give full ADC reading
* Red and blue LED indicator on-board
* 1 x EUSART (TX-RX pair)
* 1 x on-board buzzer-sounder
* On the edge connector more signals are unallocated and available, e.g. opto-isolated inputs, N-channel open-collector output.

Co-routine cooperative multitasking

* cocoOS RTOS kernel source code is included in this distro
* use of event, semaphore and priority is kept to zero whenever possible
* The cocoOS license agreement text is source code embedded. The text will flash-write to the target device, intended by design to reside in Flash Program Memory. In the end user application, sending “eula\n” to EUSART1 to obtain the text.

The code modules

edgeDetect uses a combination of comparator, DAC and timer to detect pulse period. The code module implemented three gears covering low end to mid-range all the way up to mega pulse per second. Subject to extensive bench-top testing during code development, performance in terms of execution speed and accuracy is excellent.

adcPulseSensing uses on-chip ADC in combination with two 16-bit timers for pulse detection. This attempt was also successful. Since the edgeDetect module demonstrated superior performance, the adcPulseSensing module is kept only for future reference.

audiovisual This module implemented algorithm that drives various AV devices such sounders, LEDs and dry contacts. It is compact as well as transportable.

i2a performs various text to binary value conversions. Standard functions such as itoa(), atoi(), sprintf() are not used for the reason that these conversions takes potentially very long execution time on a 8-bit microcontroller. i2a splits these long computation into shorter execution segment. Coupled with cocoOs RTOS, the combination achieves maximum overall responsiveness and performance.

alarm is very simple and elegant. Of all modules, this one is most tunable by the user.

Other resources

There are inputs and outputs that can be allocated during future expansion and adaptation. The external EEPROM is not yet used. With an RTC, this device can be used to log events with proper date-time stamping.

Low power sleep is not yet implemented. It is expected to be implemented using a “sleep task”.

**Reference**:

<https://www.steel-mate.co.uk/acatalog/Analogue-Speed-Pulse-Interface-50.html>

Steelmate Automotive advertises the “Analogue Speed Pulse Interface” at £23.40 per unit suggesting our *PB Control Box - SPA* is commercially viable.