UESB

# Intro

# Hardware

* Attiny816
* LV8548

See “Datasheets”

## Pin Mapping

Table 1: Motor Pin Mapping

| Motor | Port | | GPIO | | PIN | |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | A | B | A | B |
| 1 | PA2 | PA1 | 15 | 14 | 18 | 17 |
| 2 | PA3 | PC3 | 16 | 13 | 19 | 15 |
| 3 | PC2 | PC1 | 12 | 11 | 14 | 13 |
| 4 | PC0 | PB0 | 10 | 9 | 12 | 11 |
| 5 | PB1 | PB2 | 8 | 7 | 10 | 9 |
| 6 | PB3 | PB4 | 6 | 5 | 8 | 7 |

Table 2: Input Pin Mapping

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Port | GPIO | PIN |
| Rcin 1 | PA7 |  |  |
| Rcin 2 | PB5 |  |  |
| Switch 1 |  |  |  |
| Switch 2 |  |  |  |
| Switch 3 |  |  |  |
| Switch 4 |  |  |  |

# Firmware

## Setup

The motor and rc in pins are set as outputs/inputs using direct port access via the PORTx struct, as per tables 1 and 2.

## Timers

### TimerA (Motor PWM)

TCA0 is a 16 bit timer. The prescaler is set to 16 so that at 16MHz the counter increments every 1us. This is then used for driving the motor PWM.

A compare interrupt (CMP0) and overflow interrupt (OVF) are enabled on the timer.

When the timer count (TCA0.CNT) reaches the value set by TCA0.SINGLE.CMP0 the TCA0\_CMP0\_vect interrupt service routine (ISR) is run. The ISR turns the current motor off.

When the timer count reaches TCA0.SINGLE.PER, the over flow ISR, TCA0\_OVF\_vect runs and the count is reset to 0. The ISR turns the current motor on.

TCA0.SINGLE.PER is set to 400 so the timer resets at 2.5kHz.

The value of TCA0.SINGLE.CMP0 is varied to change the duty cycle of PWM signal. A larger value means the motor will stay on longer during the duty cycle.

### TimerB (Rcin Timestamp)

TCB0 is set to be driven directly from TCA0, so also runs at 1MHz. It is 16 bit so can count up to 65536us, plenty to measure the pulse width from an rc servo signal.

It is used purely as a timestamp to measure the pulse width between rising/falling edges on the 2 rc input pins.

## Pin Interrupts

Pin change interrupts are connected to the 2 Rcin pins (PORTA\_PORT\_vect and PORTB\_PORT\_vect). During each of these interrupts the pin state is read to check if it was a rising or falling edge, and the timestamp taken from TCB0 to measure the pulse length. The info is stored in *rcin.*

## Main Loop

### Switch Check

Switch 2 is read to switch between motor selection modes. If the switch is high (down), the motor is selected according the the pulse width on Rcin1. If it is low (up), the motor can be incremented (and decremented) by a 2 position (or 3 position) switch.

### Motor **Selection**

Checks if the pulse width on rcin1 has changed and selects the motor according to the current mode of operation

### Motor Update

Checks if the pulse width on rcin2 has changed and updates the TCA0.SINGLE.CMP0 register and motor direction according to the current settings, to change the duty cycle of the

## **Set** Motor Pin

Changes the outputs pins that connect to the motor drivers in order to set the *motor* to the required *direction*, and either on or off depending on *state*.

Direct port manipulation and function is inlined to improve performance when called inside an ISR.

# Uploading

The UESB-main.ino sketch needs the [megaTinyCore](https://github.com/SpenceKonde/megaTinyCore).

## Settings/Fuses

* Board: 816 (without optiboot)
* Clock: 16MHz (internal)
* millis()/micros disabled

Rest of settings as default are fine