# ACDC DAQ Programers Manual

Jonathan Eisch, Miles Lucas, Eric Oberla and Others

## Contents



## 1 Introduction

This document is intended to document the commands sent to the ACC and ACDC cards.

It includes source listings of the source and destinations for each command as they were when this documentation was written. New implementations may have been written since, but that is outside the scope of this document.

# 2 Commands

## 2.1 Template

### 2.1.1 Bit Fields

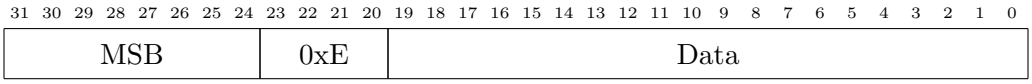


Figure 1: Command 0xE bit fields

| Bit Range | Name | Description |
|---|---|---|
| 31-24 | MSB | This is really the most significant byte |
| 23-20 | **0xE** | Command marker |
| 19-0 | Data | All of the data. |

### 2.1.2 Source

**acdc-daq:src/DAQinstruction.cpp**

```
void SuMo::software_trigger_slaveDevice(unsigned int
    ↪ SOFT_TRIG_MASK, bool set_bin, unsigned int bin)
{
    usb2.createHandles();
    //usb.sendData((unsigned int)0x000E0000);//software trigger
    const unsigned int hi_cmd = 0x000E0000;
    unsigned int send_word = hi_cmd | SOFT_TRIG_MASK | set_bin <<
        ↪ 4 | bin << 5;
    usb2.sendData(send_word);
    //printf("sent software trigger\n");
    usb2.freeHandles();
}
```

### 2.1.3 Destination

**ACDC-Firmware:SRC/dff_async_rst.vhd**

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.numeric_std.all;
```

```
entity dff_async_rst is
  port( d, clk, reset: in std_logic;
      q: out std_logic);
end dff_async_rst;
architecture behav of dff_async_rst is
  begin
  process(clk,reset)
  begin
  if (reset='1') then
    q <= '0';
  elsif rising_edge(clk) then
    q <= d;
  end if;
  end process;
end behav;
```