



Fuel Injector Driver

OPERATION MANUAL



Designed and manufactured by:
Michigan Custom Machines, Inc.
22750 Heslip Dr.
Novi, Michigan 48375

www.mcm1.com



Table of Contents

| | | |
|--------|---|----|
| 1 | Introduction | 4 |
| 2 | Variations and Serial Numbers..... | 5 |
| 2.1 | Control Options | 6 |
| 2.1.1 | XciteE Companion (Method 1)..... | 6 |
| 2.1.2 | Custom Software (Method 2)..... | 6 |
| 2.1.3 | Ethernet IP (Method 3) | 7 |
| 2.1.4 | Hardwired Input Selection of Firing Sets & Enable (Method 4)..... | 7 |
| 3 | Modes and Variable Descriptions..... | 7 |
| 3.1 | Timing Modes..... | 7 |
| 3.1.1 | OPR Synchronous Mode | 7 |
| 3.1.2 | OPR+WHEEL Synchronous Mode | 8 |
| 3.2 | Firing Modes | 8 |
| 3.3 | Normal Mode Operation..... | 8 |
| 3.4 | BIP Mode Operation..... | 10 |
| 3.5 | Building a waveform..... | 11 |
| 3.6 | Static Fire Mode | 11 |
| 3.7 | Storing/Recalling Firing Sets..... | 11 |
| 3.8 | Control Variables..... | 13 |
| 3.9 | Diagnostic Error Codes | 16 |
| 3.10 | Sample Programming | 18 |
| 3.10.1 | High-Voltage, Dual-shot injector:..... | 18 |
| 3.10.2 | Low-Voltage, BIP injector: | 20 |
| 4 | Serial Interface | 22 |
| 4.1 | Standards Compliance..... | 22 |
| 4.2 | Character Format and Bit Timing | 22 |
| 4.3 | Message Format | 22 |
| 4.3.1 | Checksum calculation detail: | 23 |
| 5 | Ethernet I/P Interface | 23 |
| 5.1 | PLC Integration | 23 |
| 5.2 | Changing the IP Address | 24 |
| 5.3 | Using Ethernet I/P and the serial interface together..... | 24 |
| 6 | Connections and Wiring | 25 |
| 6.1.1 | Isolation and Grounding | 25 |
| 6.1.2 | Sample Wiring Diagram..... | 26 |
| 6.1.3 | Power Connector Pin Descriptions | 26 |
| 6.1.4 | Injector Connector Pin Descriptions | 27 |
| 6.1.5 | External I/O Connector | 27 |
| 6.1.6 | RJ12 Serial Connector | 28 |
| 6.1.7 | RJ45 Field IO Connector | 29 |
| 7 | Troubleshooting..... | 30 |
| 8 | Warranty..... | 31 |
| 9 | Appendix A – Physical Data | 31 |
| 10 | Appendix B – Specifications..... | 32 |



11 Appendix C Front and Rear Labels 32



1 Introduction

The XciteE Driver is used to fire a wide variety of solenoid-based fuel injectors in a controlled test environment. It has high output power and high precision control of current which is suitable for both Diesel and Direct-Injection Gasoline injectors. An injector may be fired asynchronously, or in sync with an external timing signal on a flywheel for example. The output voltage is programmable, along with a fully-programmable output current waveform using “phases”. By defining these phases, split injection or multi-tiered cycles are possible. Finally, there is support for “BIP” injectors, which require an “event-driven” cycle typical of older, lower voltage injectors.

Care was taken to allow several forms of integration with a variety of control systems and use cases:

- Legacy Serial Communication ideal for PC control
- Ethernet I/P Communication ideal for PLC control (Optional)
- Ability to select stored “presets” from the software or the electrical connector
- Combination of all of the above

Our robust design allows detection of open injectors as well as shorted injectors and the driver will protect itself in these cases, not incurring damage.



2 Variations and Serial Numbers

The XciteE Driver Modules contain a 4 digit serial number, along with an optional suffix. The serial number ranges are:

| Serial Number Range | Meaning |
|---------------------|---|
| 1000-1999 | Earlier versions of this driver. Formerly called the PFIM Driver |
| 2000 and up | XciteE Driver |

The suffix determines the functional configuration, per the following table:

| Suffix for Serial Numbers ≥ 2000 | Meaning |
|---------------------------------------|---------------------------------|
| [No Suffix] | Standard XciteE w/o Ethernet IP |
| -E suffix | Standard XciteE w/ Ethernet IP |
| [other] | XciteE with custom firmware |



2.1 Control Options

The XciteE is easily controlled or setup via its RJ12 Main UART serial interface on the X4 Connector, however there are 4 options for control:

2.1.1 XciteE Companion (Method 1)

The XciteE Companion software runs on a PC and allows the user to program and control the XciteE driver. It supports up to 20 injector phases (including injecting events with programmed current levels, and pauses between injections). This software is free and comes with the XciteE driver.

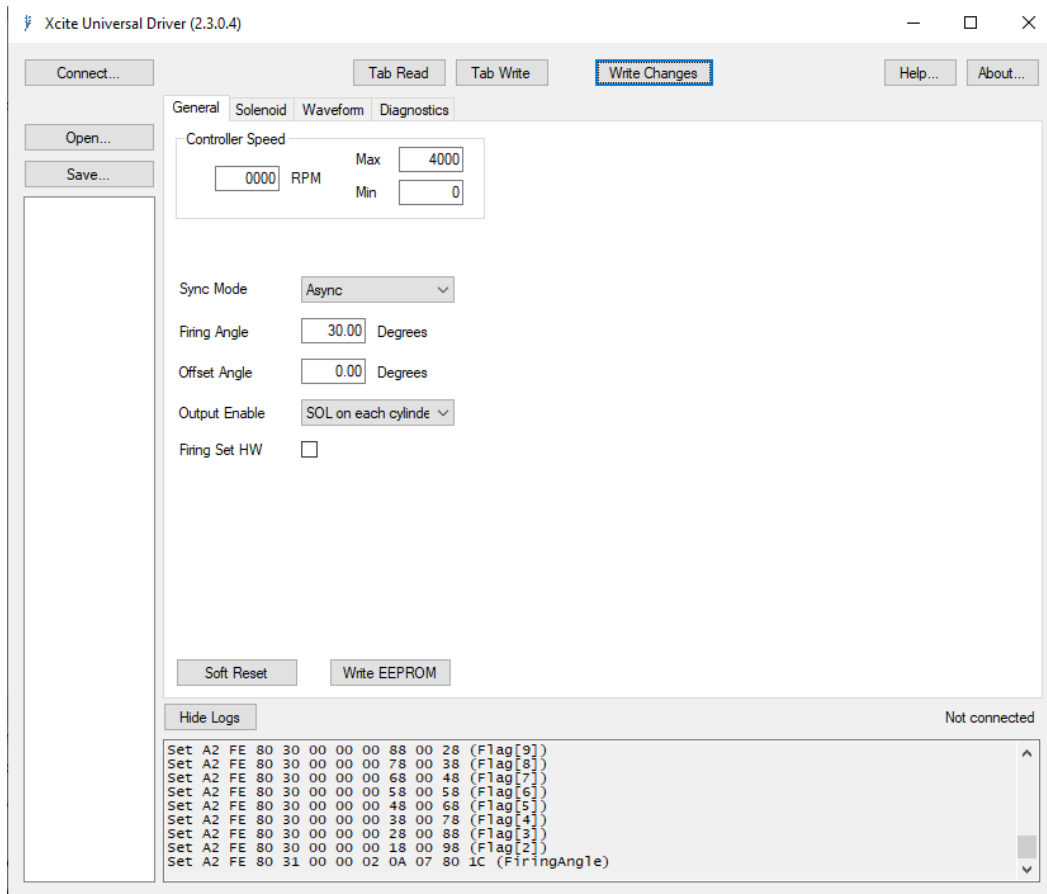


Figure 2.1.1-1 Xcite Companion Software

2.1.2 Custom Software (Method 2)

The serial protocol may be used to control the XciteE Driver at a low level. The XciteE Driver provides an address-based set of parameters which are available for reading or writing. The protocol is documented later in this document, as are all of the variables.



In either case, some parameters can be stored in EEPROM, and will become defaults at startup. Others must be downloaded into the XciteE Driver before they take effect. This is documented in the section on **Variable Descriptions**.

2.1.3 Ethernet IP (Method 3)

There is a port on the rear of the unit that optionally supports the Ethernet IP protocol. All units have this port, but the -E suffix at the end of the serial number indicates that a daughter board is installed to support this form of communication. If this option is purchased, any PLC or device that supports the Ethernet IP protocol can communicate with the driver. For convenience, we provide at no charge, EDS files, supporting files and programming examples for some PLCs. All of the variables and live readings that are available in XciteE Companion are available through this port as well.

2.1.4 Hardwired Input Selection of Firing Sets & Enable (Method 4)

The XciteE Driver may be programmed for up to four distinct firing sets using one of the other three methods. Once programmed, individual firing sets may be selected based on the status of digital selection inputs. The XciteE Driver may be enabled or disabled based on the status of the hardware Enable Bit. This allows operation and setup & waveform selection without connection to the RJ12 Main UART serial interface (using pre-stored setups). This is ideal for the case where there is no PC or PLC control and selection is to be made with a selector or push button.

3 Modes and Variable Descriptions

3.1 Timing Modes

Injector firing may be **synchronous** or **asynchronous**. Asynchronous firing simply repeats at a given rate and does not require an external input. Synchronous firing requires a “once-per-rev” signal be fed to the INPUTA pin. Firing can then take place at any angle with respect to the “once-per-rev” signal. Firing angle is determined by the FIRING_ANGLE and OFFSET_ANGLE parameters. There is also a 36-tooth wheel option for synchronization, which will synchronize with a 36-tooth wheel with one tooth missing (37-1) to represent the OPR. At additional cost, custom firmware can be developed to synchronize to a different pattern.

3.1.1 OPR Synchronous Mode

In OPR Synchronous mode, Input “INA” on the IO connector serves as a “Once per Rev” input. The driver uses this input to determine the wheel speed and it bases the firing angle setting and offset with respect to this point. Since there is only a single reference point, angles are determined by a very accurate time from this point. This time is recalculated with every revolution, however they will be



firing angle error during acceleration and deceleration. If you are spinning your cam at a uniform rate, this method will work well.

3.1.2 OPR+WHEEL Synchronous Mode

This mode is similar to OPR mode, however it uses an additional input “INB” on the io connector for a regular pattern used to control angle. This can be a toothed wheel with a pickup or a disk with slots in it, or an industrial encoder. The driver will automatically determine how many teeth are on your wheel every time you put it in OPR+WHEEL mode. The first time the wheel spins, it will take three revolutions to learn the wheel pattern but once learned, it will not need to do this again, while it remains powered.

This mode is angle based and accurately controls the firing angle of the injector by keeping track of the pulses. Where firing happens between pulses, the driver will interpolate with time, so for example if a 360 tooth wheel is used, there is 1 degree per tooth and if it were desired to fire at 25.5 degrees, the remaining 0.5 degrees would be timed out. The more teeth on the wheel, the more accurate the angle will be. There are limitations to maximum RPM when using wheel decoding. With a 1024 teal wheel, the maximum speed that the driver can keep pace with is 2500 RPM. So if it was desired to rotate at 5000 RPM, then a wheel will less than 512 teeth would be required. In this mode, angle will track with changes in speed which can be desirable if the application requires this.

3.2 Firing Modes

The XciteE Driver can operate in one of two modes, **Normal** and **BIP** mode. They are described separately in the following sections, but if you are not familiar with these waveforms, you can use still easily determine which waveform to use for your application. All GDI Gasoline injectors use Normal mode. Any diesel injector that has a solenoid that is activated with 24 VDC or less is likely using a BIP strategy, otherwise it is a normal strategy.

3.3 Normal Mode Operation

The injector is fired for a particular number of timed phases up to 20 maximum. Each phase has a target current level, target chop current level, duration and the option of using boost voltage or battery voltage. Waveforms will be discussed more in section 3.5. At the beginning of the phase, the output voltage is turned on to reach the intended target current. When the target current is reached, the XciteE Driver will switch on and off the output voltage to regulate the current to within the chop current level, if possible. When the duration is expired, the next phase begins.

A phase with current level 0 and a duration will allow for split injection cycles, creating a pause between phases.



During the cycle, the output voltage comes from the BOOST_VOLTAGE parameter. The boost supply is programmable, from the input voltage (Battery) up to 110VDC. This can be used to change the initial injector current rise rate.

When the current is dropping to zero, (if ZENER_ALWAYS=0), the zener circuit limits the flyback voltage to $V_{BOOST_VOLTAGE} + ZENER_VOLTAGE + 10$. This can be used to control the final injector current fall rate. (if ZENER_ALWAYS=1, this will affect all current fall rates)



3.4 BIP Mode Operation

This mode came about from the early EUI days of diesel. It allows low-voltage injectors to have more repeatability due to the slower, indeterministic pull-in rate. The BIP cycle compensates for variable injector pull in rate by monitoring the current on the injector solenoid. When the injector solenoid has completely pulled in the armature, this is detected and the second phase begins. This scheme minimizes the varied amount of fuel that might be discharged based on the speed of the pull-in on the first phase. Split injection is not supported in BIP mode. (Only phases D1 and D2 may be used.) In addition, the internal boost supply is disabled in BIP mode, and the supplied voltage is normally the VBATT input voltage.

D1_CHOP_AMPLITUDE should be set to zero. Pull in time begins with VBATT applied. When D1_CURRENT is reached, the voltage supply switched from VBATT to VBIP, which is a programmable low-voltage supply whose voltage is set via BIP_VOLTAGE*. The current trace generally hits a peak around this time, as the supplies switch. (The current peak may overshoot the D1_CURRENT threshold in BIP mode, based on the supply switching time. This parameter value should be tuned as needed to reach the desired peak) The XciteE Driver now waits for a “positive curvature point” in the current waveform. (The sensitivity of this process is controlled by BIP_THRESHOLD.) If it detects this point (BIP), it switches back to VBATT and runs as usual for the D2 phase. If it doesn’t detect BIP, it will timeout this phase based on the duration of the first phase. That’s is, in BIP mode the length of the first phase should be longer than the time it will take for the injector to pull in as it is used as a timeout.

Also, in synchronous mode, the point of BIP detection is used for another purpose. As far as the firing angle is concerned, this point is used as the firing time, as opposed to the start of current point in Normal mode. There are two BIP modes where one is synchronized with SOL and the other is synchronized with BIP. In the case of “Synchronize with BIP”, the XciteE Driver adjusts the angle such that the firing always starts with BIP detection. If “Synchronize with SOL”, the waveform will be synchronized with the start of logic, where the waveform starts.

3.5 Building a waveform

The shape of the waveform is defined by “phases”. The XciteE Driver can handle up to 20 total phases. These phases can be used to build a simple pull-in and hold-in waveform, a complex multi-tiered waveform or a several shot waveform. Each phase consists of four components; a current level, duration, chop amplitude and voltage. Three of the four components are labeled in the diagram below for phase 1, the voltage cannot be labelled as it is not identifiable in this illustration. In the cases where there is a gap between two shots, a phase with a duration, but zero current and zero amplitude are used such as in phase 3 shown below. The last phase (if all 20 are used) is assumed to be zero. If a phase has a duration of zero, it is treated as if it didn’t exist.

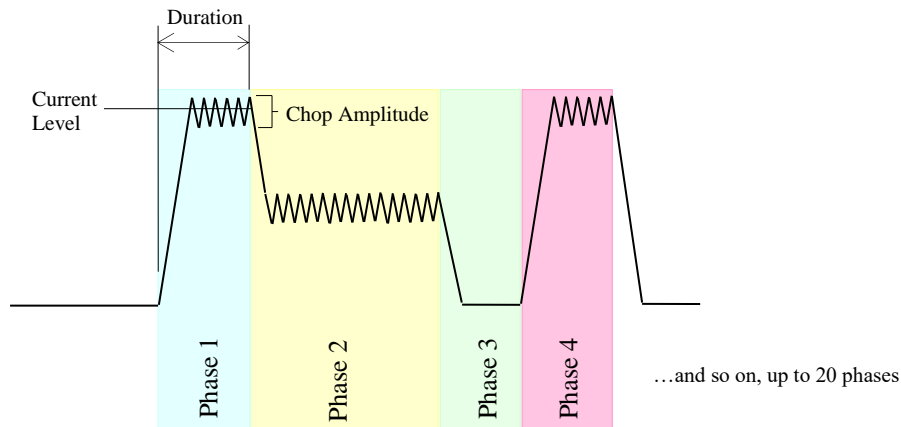


Figure 3.1.2-1 Phase Diagram

3.6 Static Fire Mode

Static fire mode will hold the injector open for up to 45 seconds. This is accomplished by commanding 1 RPM to the driver. When this special mode is enabled, the first two phases will only be used, where the second phase will hold until the RPM command is changed back to zero or the 45 seconds has elapsed. In the second phase the boost voltage is forced off and battery voltage is used to hold the injector open at the specified current for the 2nd phase. **Be aware of your injector and how much current it can handle as you may burn out the injector solenoid in this mode.** If the 45 second timeout elapses, an ERROR_STATIC_TIMEOUT fault will be set and will clear on the next valid firing operation. Note that this mode works in asynchronous and synchronous modes.

3.7 Storing/Recalling Firing Sets

The XciteE driver has the ability to support the selection of firing sets. A firing set is a complete waveform that can be recalled from digital inputs on the IO connector or through the software command interface. In order to use a firing



set, one must first be stored. If this is the first time you are using the driver, the stored waveforms will be blank.

To Store a waveform, first set the waveform parameters to your liking. If you are using XciteE Companion, there is a “Store Firing Set” button that easily allows you to make this selection. If the action is being performed by custom software or through the Ethernet IP port, the following parameters need to be set accordingly.

- `FIRING_SET_STORE_RECALL_SELECTION = X`
Where X is 0,1,2 or 3, which represents the firing set number that you wish to write to
- `FIRING_SET_STORE_RECALL_ACTION = 1`

Upon successful execution, the currently programmed waveform will be written to Non Volatile memory in the location specified. If an invalid location is specified, the command will fail.

To Recall a waveform from non-volatile memory, it is a similar procedure. In XciteE Companion, you select the set that you wish to recall and the active waveform represents the firing set selected. When performing a recall from custom software or the Ethernet IP interface you would set the following parameters accordingly.

- `FIRING_SET_STORE_RECALL_SELECTION = X`
Where X is 0,1,2 or 3, which represents the firing set number that you wish to write to
- `FIRING_SET_STORE_RECALL_ACTION = 2`

Upon successful execution, the selected firing set will be recalled from non-volatile memory into the active waveform. If an invalid location is specified, the command will fail.

The `FIRING_SET_HARDWARE_SELECT_ENABLE` parameter determines if the firing set is chosen from the X5 IO Connector.

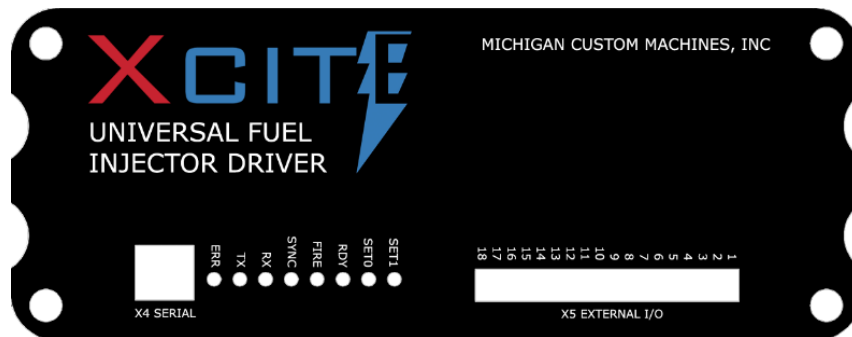


Figure 3.7-3.1.2-1 Hardware Firing set selection



The following table shows the effect of the signals on the IO connector, pins 9 and 10 for set selection. Note that a “1” is a positive signal and a “0” is ground. Pins 17 and 18 are ground reference while pin 6 is a 5V reference if needed. Refer to section 6.1.5 External I/O Connector, for more wiring information.

| Pin 9 INSET0 | Pin 10 INSET1 | Set Selection |
|-----------------|------------------|---------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 3 |

When a hardware set has been selected the SET0 and SET1 LEDs on the front panel will illuminate accordingly. **Note that pin 8 on the IO connector (INENABLE) is used in this mode to turn the injector on and off.** It may be desirable to enable the injector only during testing for example and when using hardware sets, the profile is recalled immediately. In the case where an asynchronous profile was recalled the injector would always fire if pin 8 was wired HIGH.

3.8 Control Variables

This section describes all of the variables that define the operation of the XciteE driver. Some of the variables are saved to non-volatile RAM, while others are stored in RAM and initialized to zero on powerup. When writing custom code to communicate via the serial port, the address and size are part of the formed message. Please see section 4 for details on the serial communication.

| Name | Address | Size | Units | Description |
|-------------------|-----------------------|------|-------|--|
| D1_CURRENT | 0x00000000 | 2 | mA | target current level |
| D1_CHOP_AMPLITUDE | 0x00000002 | 2 | mA | target current “dither” level |
| D1_DURATION | 0x00000004 | 4 | uSec | time for this phase |
| D1_VBOOST | 0x00000008 | 1 | 0,1,3 | Use Global, Force OFF, Force ON |
| D2_CURRENT | 0x00000010 | 2 | mA | target current level |
| D2_CHOP_AMPLITUDE | 0x00000012 | 2 | mA | target current “dither” level |
| D2_DURATION | 0x00000014 | 4 | uSec | time for this phase |
| D2_VBOOST | 0x00000018 | 1 | 0,1,3 | Use Global, Force OFF, Force ON |
| D3...D20 | 0x00000020-0x00000148 | ... | ... | (...etc., through D20) |
| RPM | 0x00000200 | 2 | None | 0=disable, >=100 enable async or sync firing |
| RPM_MEASURED | 0x00000202 | 2 | None | read only |
| SYNC_MODE | 0x00000205 | 1 | 0,1 | 0 = async, 1 = sync to one pulse per rev |



| | | | | |
|---------------|------------|---|---------------|--|
| BOOST_VOLTAGE | 0x00000206 | 2 | V | controls risetime along with injector |
| ZENER_VOLTAGE | 0x00000208 | 2 | V | controls fall-time along with injector, set to about 10V higher than Vzener |
| FIRING_ANGLE | 0x0000020a | 2 | +1/64 degrees | In sync mode, the formula for firing time is (FIRING_ANGLE*64 – OFFSET_ANGLE*64) degrees after the INPUTA rising edge |
| OFFSET_ANGLE | 0x0000020c | 2 | +1/64 degrees | Cam angle when the 1/rev happens. Resultant firing angle from 1/rev will be Firing Angle minus Firing Angle Offset. See Section 3.1 in manual |
| VERSION | 0x0000020e | 2 | code | read only |
| ZENER_ALWAYS | 0x00000217 | 1 | 0-1 | 0 = engage zener only when dropping to 0 Amps. 1 = engage zener when dropping to any threshold current |
| RPM_MAX | 0x00000218 | 2 | RPM | limits for input filtering |
| RPM_MIN | 0x0000021a | 2 | RPM | limits for input filtering |
| BUILD_VERSION | 0x00000222 | 2 | code | read only |
| EE_WRITE | 0x00000224 | 1 | any | writing a 1 will store current values of 0x00000204-0x000002ff to non-volatile memory |
| SOFT_RESET | 0x00000225 | 1 | any | writing a 1 will do a soft reset |
| ERROR_CODE | 0x00000226 | 1 | code | 0 = no error |
| BIP_MODE | 0x00000238 | 1 | 0-2 | 0 = Normal mode (boost supply enabled, synchronous firing time is “start of current”) 1 = BIP mode (boost supply disabled, synchronous firing time is “BIP detected”) 2 = BIP mode (boost supply disabled, synchronous firing time is SOL) |
| BIP_THRESH | 0x0000023a | 2 | mA | Sensitivity level adjustment for BIP triggering |
| BIP_VOLTAGE | 0x0000023e | 2 | volts | Set voltage for BIP mode |



| | | | | |
|-----------------------------------|-----------|---|-----------|---|
| FIRING_SET_HARDWARE_SELECT_ENABLE | 0x0000250 | 1 | 0,1 | 0 = No firing set will be used 1 = Firing set will be determined by digital inputs on I/O Connector |
| FIRING_SET_STORE_RECALL_SELECTION | 0x0000252 | 1 | 0,1,2,3 | Firing set selected |
| FIRING_SET_STORE_RECALL_ACTION | 0x0000253 | 1 | 0,1,2 | 0 = No Action 1 = Write Selected set 2 = Read Selected Set |
| ERROR_MASK_0 | 0x0000300 | 2 | flag bits | error enable bits for codes 0x00-0x0f |
| ERROR_MASK_1 | 0x0000302 | 2 | flag bits | error enable bits for codes 0x10-0x1f |
| ERROR_MASK_2 | 0x0000304 | 2 | flag bits | error enable bits for codes 0x20-0x2f |
| ERROR_MASK_3 | 0x0000306 | 2 | flag bits | error enable bits for codes 0x30-0x3f |
| ERROR_MASK_4 | 0x0000308 | 2 | flag bits | error enable bits for codes 0x40-0x4f |
| ERROR_MASK_5 | 0x000030a | 2 | flag bits | error enable bits for codes 0x50-0x5f |
| ERROR_MASK_6 | 0x000030c | 2 | flag bits | error enable bits for codes 0x60-0x6f |
| ERROR_MASK_7 | 0x000030e | 2 | flag bits | error enable bits for codes 0x70-0x7f |
| BAUD_RATE | 0x0000350 | 1 | 5-12 | Baud Rate selection Default = 115,200 5= 4800 6= 9600 7= 14400 8= 19200 9= 38400 10= 57600 11= 115200 12= 230400 |
| DIAG_VBOOST_MAX | 0x0000400 | 2 | volts*100 | Maximum Internal Voltage |
| DIAG_ISENSE_MAX | 0x0000402 | 2 | mAmps | Maximum Internal Current |
| DIAG_ISENSE_MIN | 0x0000404 | 2 | mAmps | Maximum Output Current when not firing |
| DIAG_VBATT_MIN | 0x0000406 | 2 | volts*100 | Minimum Supply Voltage |
| DIAG_VBATT_MAX | 0x0000408 | 2 | volts*100 | Maximum Supply Voltage |
| DIAG_VBATT_HYST | 0x000040a | 2 | volts*100 | Hysteresis for self-clearing VBATT_MAX or VBATT_MIN errors |
| DIAG_RISETIME_MIN | 0x000040c | 2 | uSec | Longest Risetime which indicates a shorted injector |
| DIAG_THROTTLE_MAX | 0x000040e | 1 | 0-20 | Used for Power Supply Limiting. This value will limit the boost supply power. |



| | | | | |
|---------------|------------|---|-------------|--|
| FIXED_SHOTS | 0x00000410 | 2 | 0-10000 | If this field is zero, normal operation. In cases where it is non zero, that number of shots will be fired and then subsequent shots will cease. Setting RPM to 0 will reset the action. Only supported through the serial port. |
| ETHIP_ADDRESS | 0x00000413 | 4 | aa.bb.cc.dd | 4 byte word (32 bits) that reflect each octet in the IP address where each octet is represented by one of the bytes in aabbccdd order. |

Figure 3.1.2-1 Message Table

3.9 Diagnostic Error Codes

The RED Error LED on the front of the XcitE driver should be off under normal operation. If an error is present, the LED will pulse a number of times for the first digit, pause and then pulse the number of times for the second digit and then finally a long pause. The cycle will then continue to repeat until the error has been corrected.

The XciteE Driver will continue to run with error codes 0x31-0x33. Critical Errors (0x11-0x1A) will cause a shutdown, and require a restart to recover from them.

| Error Code (hex) | Description | ERROR_MASK address & bit for enable/disable |
|------------------|---|---|
| 0x11 | Internal Overvoltage | ERROR_CODE_1, bit 1 |
| 0x12 | Internal Overcurrent | ERROR_CODE_1, bit 2 |
| 0x13 | External Overcurrent | ERROR_CODE_1, bit 3 |
| 0x21 | External Overvoltage | ERROR_CODE_2, bit 1 |
| 0x22 | External Undercurrent | ERROR_CODE_2, bit 2 |
| 0x23 | External Undervoltage | ERROR_CODE_2, bit 3 |
| 0x24 | Wheel Tooth Mismatch (Sync+WHL Mode only) | ERROR_CODE_2, Bit 4 |
| 0x31 | Injector Shorted | ERROR_CODE_3, bit 1 |
| 0x32 | Injector Open | ERROR_CODE_3, bit 2 |
| 0x33 | No BIP | ERROR_CODE_3, bit 3 |



| | | |
|------|-----------------------|---------------------|
| 0x34 | Overspeed (sync only) | ERROR_CODE_3, bit 4 |
| 0x35 | Static Timeout | ERROR_CODE_3, bit 5 |

Diagnostic Codes



Notes on error mask values:

There is an algorithmic way to figure out the mapping.
The ERROR_MASK_N word affected will be ERROR_CODE/16
The actual enable bit will be the remainder of ERROR_CODE/16

For example, if you take ERROR_CODE = 105 (Injector 1 Open)

$105/16 = 6$ with a remainder of 9

So, the enable bit is ERROR_MASK_6, bit 9.

To disable this error code, change ERROR_MASK_6 from 65535 to:

$65535 - 2^9 = 65535 - 512 = 65023$

As always, it is a good idea not to disable things unless you know it's safe.
We provide this ability in the event that the driver is keeping you from doing something that is unorthodox but intentional, however the driver has a wide range of flexibility that should not limit the end user.

Disabling the diagnostics may result in damage to the XciteE Driver and may void the warranty.

3.10 Sample Programming

We will give two examples of setting up the XciteE Driver.

3.10.1 High-Voltage, Dual-shot injector:

In this case, our requirements are:

- Synchronous firing
- Firing angle = -20 degrees before “once-per-rev”
- Voltage = 75V
- Shot 1 Pull-in current = 7 Amps
- Shot 1 Hold-in current = 5 Amps



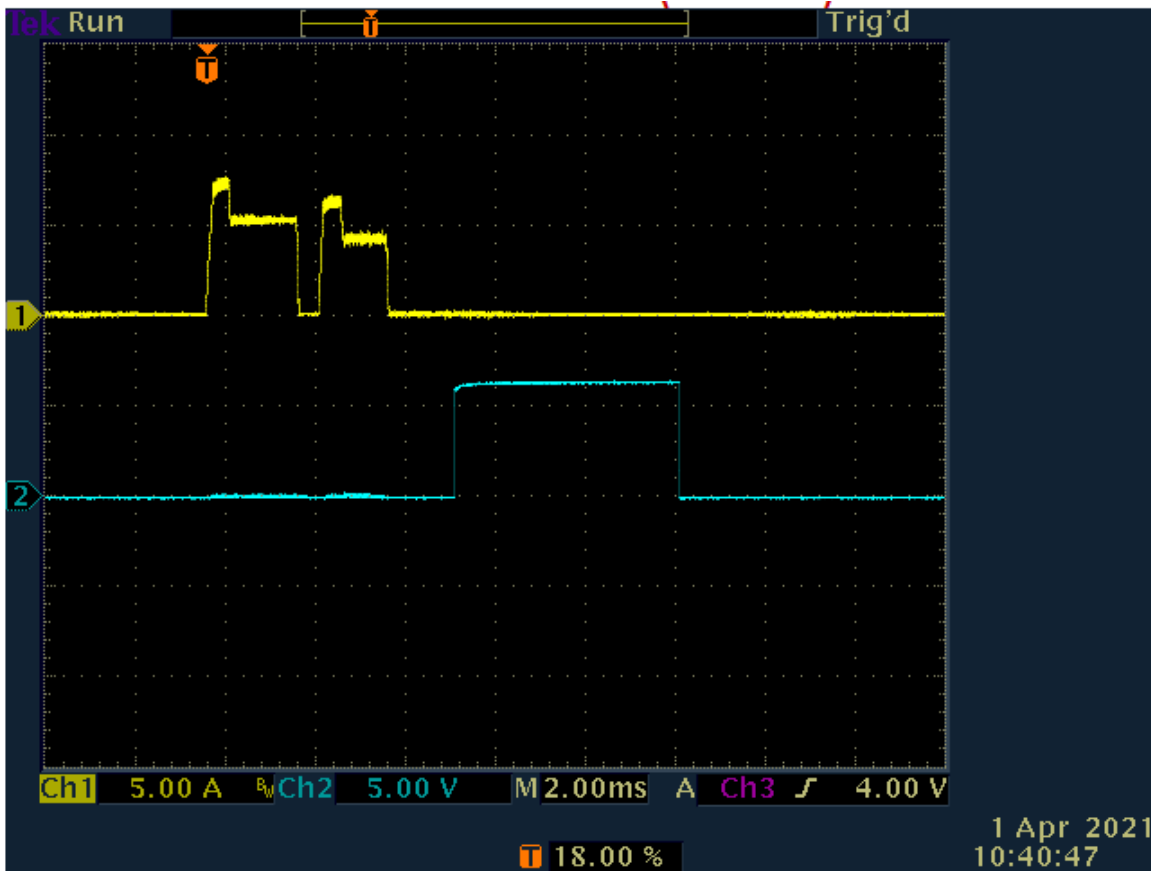
- Shot 1 Pull-in duration = 500us
- Shot 1 total duration = 2000us
- Delay between Shot 1 and Shot 2 = 500us
- Shot 2 Pull-in current = 6 Amps
- Shot 2 Hold-in current = 4 Amps
- Shot 2 Pull-in duration = 500us
- Shot 2 total duration = 1500us

For this case, we would set the following variables:

```
SYNC_MODE = 1
FIRING_ANGLE = -20*64 = -1280
BOOST_VOLTAGE = 75
D1_CURRENT = 7000
D1_DURATION = 500
D2_CURRENT = 5000
D2_DURATION = 2000-500 = 1500
D3_DURATION = 500
D4_CURRENT = 6000
D4_DURATION = 500
D5_CURRENT = 4000
D5_DURATION = 1500-500 = 1000
```

We would also set `ZENER_VOLTAGE = BOOST_VOLTAGE`, unless we have a spec for it, to allow the valve to close quickly and repeatably.

Finally, we set `RPM = 100` to enable firing
`RPM = 0` to disable firing



Ch1 Injector Current
Ch2 1/rev signal

3.10.2 Low-Voltage, BIP injector:

In this case, our requirements are:

- Asynchronous firing
- Battery Voltage = 18V
- BIP Voltage = 5V
- Pull-in current threshold = 10 Amps
- Hold-in current = 12 Amps
- Injection duration = 2000us

For this case, we would set the following variables:

SYNC_MODE = 0
BOOST_VOLTAGE = 0

D1_CURRENT = 8500
(We set this to a value lower than 10A, to look for the current peak to be 10A)



D1_DURATION (BIP_TIMEOUT) = 3000

This should be set to the timeout value, at least > 2x the longest expected value

D2_CURRENT = 12000

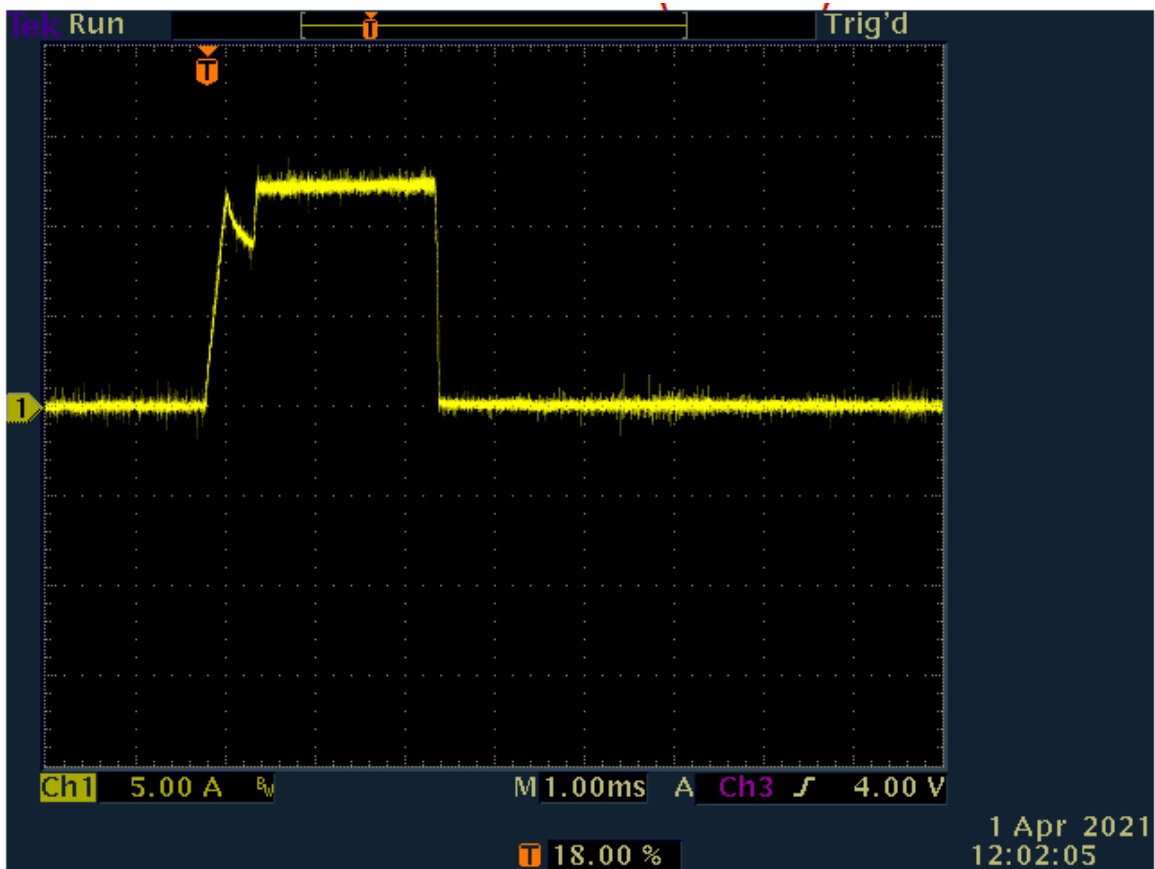
D2_DURATION = 2000

We also require setting BIP_THRESHOLD to a value which provides a stable waveform for this configuration. (Again, tweak as needed while watching the current waveform)

ZENER_VOLTAGE = 18V (> VBATT) to insure timely valve closure

Finally, we set RPM = 200 to enable firing at 200 RPM (any desired RPM)

RPM = 0 to disable firing.



Ch.1 Injector Current



4 Serial Interface

4.1 Standards Compliance

The XciteE Driver communicates primarily over the RJ12 Main UART Serial Connector via a simple RS-232 serial link. Bit timing, character format, and message format is compatible with section 6 of the SAE J1708 standards document. In addition, the XciteE Driver uses codes and manufacturer-specific data fields, which are compatible with the message definitions of SAE J1587.

4.2 Character Format and Bit Timing

9600 baud, 8 bits, no parity, 1 stop bit.

The protocol consists of read and write requests by the host, with positive reply (acknowledgement) from the XciteE Driver. Lack of a positive reply indicates that the message was not received, or contained an invalid address or length field.

It is also possible that data sent with a write request may be considered “out of range” for the given address. In this case, you will get a reply, but the data in the reply will be unchanged by the write. This is another good thing to check for in the reply.

4.3 Message Format

| Byte number(s) | Purpose | Typical Hex Value(s) |
|----------------|------------------------------|--|
| 1 | Sender ID | 0xa2 (HOST), 0x80* (XciteE Driver) |
| 2 | Escape Code | 0xfe |
| 3 | Receiver ID | 0x80* (XciteE Driver), 0xa2 (HOST) |
| 4 | Message Type and Data Length | Upper nibble is the message type: 0x2X = Read Request 0x3X = Write Request 0x4X = Acknowledgement Lower nibble is the data size: 0xXN = N+1 bytes Examples: 0xX0 = 1 byte 0xX1 = 2 bytes 0xX3 = 4 bytes |
| 5 through 8 | Address | Command address |
| 9 through 9+N | Data | Command data |
| 9+N+1 | Checksum | Two's Complement checksum byte of the message |

Figure 3.10.2-1 Serial Message Format



Typical Transaction:

Note: Actual bytes are binary – not ascii hex

(write to address 00000000, data = 4E20)

sending write request... A2FE8031000000004E2041
write response OK... 80FEA241000000004E2031

(read from address 00000000)

sending read request... A2FE802100000000BF
read response OK... 80FEA241000000004E2031

4.3.1 Checksum calculation detail:

To calculate the single-byte checksum...

- 1) Sum all bytes in the message
- 2) Take the least significant byte
- 3) Invert all bits and add one

Notes on implementation details:

The checksum system defined by J1708 is typical practice. For additional communication robustness, It is possible to validate other response data fields as well.

The XciteE Driver processes one message at a time. One should wait for a response from the current request before issuing another request. The response timeout value is unspecified, but a recommended value is 500ms.

5 Ethernet I/P Interface

5.1 PLC Integration

The XcitE driver, if equipped, is able to communicate as an Ethernet device over Ethernet IP with most PLC's that support this protocol. The benefit of this approach is that the communication is fast and guaranteed. In the case of serial communication, the command would have to be resent if there was a response timeout. Included with the driver and also available on our website are support files for this integration. The .EDS file defines the module and can be added using the Rockwell Automation hardware installation tool if you are using Allen Bradley products. We have included examples of AOI's (Add on instructions) and UDT's



(User defined types) that can be imported and used right away. The AOI's will allow scaling of the raw data to human readable parameters so that the PLC programmer does not have to deal with that intermediate step. Simply there are two tables that the PLC will have once these components have been imported; an input and an output table of parameters. The parameters are already named appropriately and whatever value the output parameter is set to automatically gets updated to the XcitE driver and the changes take place immediately. The input parameters are exactly the same as the output parameters. In cases where a parameter is read only, the output parameters are ignored and the valid data is in the input parameter. For normal read/write parameters, the input values will be set to match in the output values when they are accepted. In the case that the driver receives a parameter that is outside of the acceptable limits or is ignored, the input parameter will reflect what is actually set in the driver. By knowing this, the PLC could surmise that if the input and output values match, the parameter is properly set, however this level of overcheck is not required for reliable operation.

5.2 Changing the IP Address

The IP address of the XcitE driver is configurable two ways.

The less preferable way is through a webserver on the Ethernet port. This can be achieved by going to the IP address that the driver is set to, (typically 192.168.1.1) through any web browser connected from a PC on the same subnet. Once on this page, there is an area where the IP address can be reconfigured. Make sure that DHCP is unchecked otherwise a new IP address will be assigned when connecting to a network or may never be assigned if no DHCP server is available.

The more preferable way so set the IP address is through the serial port using the XcitE companion program. On the general tab, if the Ethernet IP option is installed in the driver, the Ethernet IP address configuration will be available. Here, the IP address is displayed and can be changed as well. Note that the PLC cannot change the IP address and that it can only be done through the serial port.

5.3 Using Ethernet I/P and the serial interface together

It is possible to use the Ethernet IP interface in the PLC but also use the serial interface with a PC at the same time. Please take the time to understand this concept because one interface will win depending on the setup.

Since the Ethernet IP data is always valid and transmitting, any commands from the serial port would be overwritten shortly after they were written by the PLC data. In this scenario, the serial port could be used to read the parameters on the XcitE driver, but would not be able to change parameters. If you wish to have the opposite scenario, where the serial port has full control but the PLC could monitor the input table, there is a way to do that. In the output command table, there is a parameter called "Data_Valid". If that parameter is a "1", which is the default, the PLC data is written to the driver as described in the previous paragraphs. If this



parameter is set to a “0”, the Driver ignores any command from the Ethernet IP device, allowing the serial port to have full control. The output table is still populated however, so that the PLC can monitor parameters in the driver. In this scenario, the PC could make a change and the PLC could watch for that change to take effect before performing a specific task, for example. If the Ethernet connection is broken, the serial port also has full control.

6 Connections and Wiring

6.1.1 Isolation and Grounding

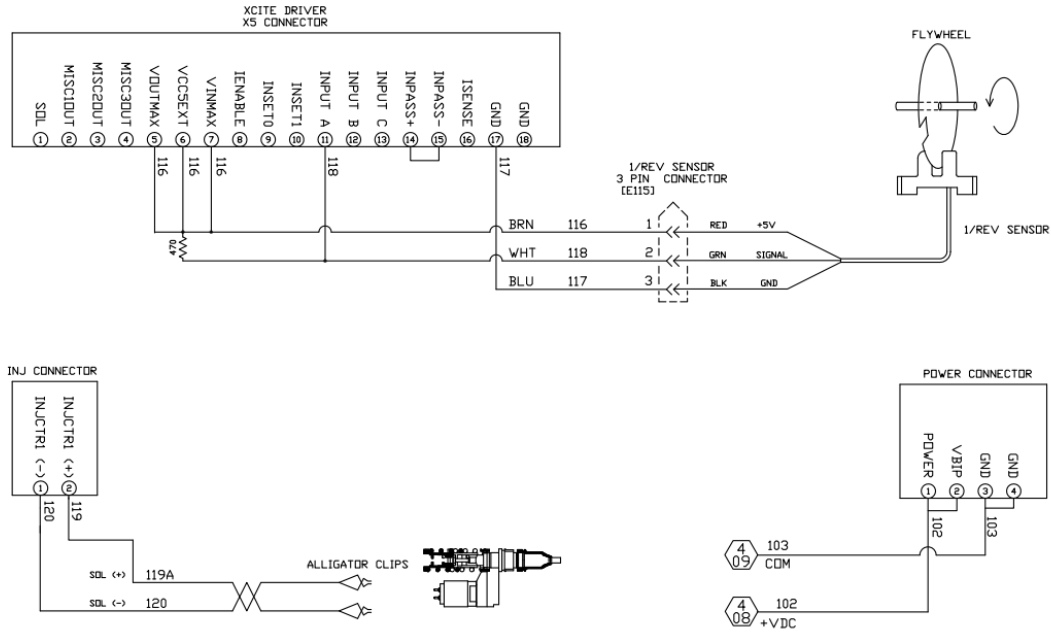
Note that the XciteE Driver External I/O is not isolated from DC power. The ground connections on the External I/O connector and the Serial connectors are allowed to be connected to the Power Ground pin inside the XciteE Driver.

For best results it is recommended to use the External I/O Ground for “signal ground”, because it will have the least noise compared to the other signals on that connector.



6.1.2 Sample Wiring Diagram

The following drawing is the recommended configurations for connecting XciteE Driver's into an injector test system.



Sample Wiring Diagram

6.1.3 Power Connector Pin Descriptions

The power connector is a 4 pin connector as show below on the back panel of the XcitE driver and is labelled X3.

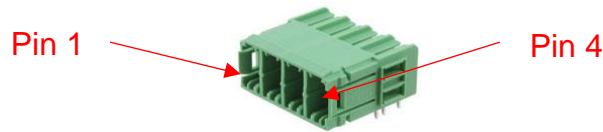


Figure 6.1.3-6.1.3-1 Power Connector (X3)

| Pin # | Pin type | Pin Name | Description |
|-------|---------------|----------|---|
| 1 | Input (Power) | VBATT | +12V to +24V DC Supply Voltage |
| 2 | Output | VBIP | BIP Voltage Supply (Normally jumped to Pin 1) |
| 3 | Ground | GND | Ground |
| 4 | Ground | GND | Ground (secondary connection point) |



6.1.4 Injector Connector Pin Descriptions

The injector connector is a 2 pin connector as show below on the back panel of the XcitE driver and is labelled X1.



Figure 6.1.4-1 Injector Connector

| Pin # | Pin Type | Pin Name | Description |
|-------|----------|----------|------------------------|
| 1 | Output | INJ1+ | Injector positive lead |
| 2 | Output | INJ1- | Injector negative lead |

6.1.5 External I/O Connector

The IO Connector is used mainly for interfacing to other control signals and diagnostics. The IO was designed so that it can function on a wide variety of voltage levels. The input and output pins can handle a DC voltage up to 24 volts and the input and output levels are individually configurable. The operation voltages levels of the digital IO on this connector are established by a reference voltage wired to pin VINMAX and VOUTMAX where VINMAX is the voltage level for digital inputs and VOUTMAX is the voltage level for digital outputs. If needed, a 5V reference voltage is available on pin 6 of the connector.



Figure 6.1.5-1 IO Connector (X5)

| Pin # | Pin Type | Pin Name | Description |
|-------|-----------------|----------|---|
| 1 | Output | SOL | “Start of Logic” Signal used to mark the activation of the injector. Can be used to gate external timing equipment |
| 2 | Output | MISC1OUT | HIGHSIDE_ON: Signal is active while the high side of the injector is active. This will be active across splits, and can be used for synchronization with other systems |
| 3 | Output | MISC2OUT | BIP: Falling edge when Pullin current threshold is reached. Rising edge when BIP detected, or BIP_TIMEOUT exceeded |
| 4 | Output | MISC3OUT | Not Used |
| 5 | Input | VOUTMAX | Scaling reference for output signals, usually tied to +5V, range 5 to 24V. Note: must be tied to a positive supply for use of outputs. |
| 6 | Reference Power | VCC5EXT | +5VDC power supply available for sensors and reference signals |
| 7 | Input | VINMAX | Scaling reference for input signals, usually tied to +5V, range 5 to 24V. Note: must be tied to a positive supply for use of inputs. |



| | | | |
|----|-----------------|----------|---|
| 8 | Input | INENABLE | Only used when using firing sets. This pin must be HIGH for the injector to fire and LOW to stop it from firing. When using firing sets, the profile is recalled and activated immediately, so it may be desirable to use this pin to enable firing when appropriate rather than all of the time. |
| 9 | Input | INSET0 | EEPROM Firing Set Selection, Bit 0 Selects one of four Pre-Configured XciteE Driver setups and waveforms based on user hardware inputs. INSETx (0 & 1): 0 & 0: Set 0 1 & 0: Set 1 0 & 1: Set 2 1 & 1: Set 3 |
| 10 | Input | INSET1 | EEPROM Firing Set Selection, Bit 1 Selects one of four Pre-Configured XciteE Driver setups and waveforms based on user hardware inputs. (See above) |
| 11 | Input | INPUTA | Main Wheel Speed Input Pin |
| 12 | Input | INPUTB | Auxiliary Wheel Speed Input Pin |
| 13 | Input | INPUTC | Auxiliary Wheel Speed Input Pin |
| 14 | Input | INPASS+ | Passive Wheel Sensor Positive Note: if not used, connect to Passive Wheel Sensor Negative. |
| 15 | Input | INPASS- | Passive Wheel Sensor Negative Note: if not used, connect to Passive Wheel Sensor Positive. |
| 16 | Output (Analog) | ISENSE | Current Sensor Output Gain = -1/15 Volts/Amp Offset = 4 Volts Note: not isolated – for reference only |
| 17 | I/O | GND | Ground reference for all signals |
| 18 | I/O | GND | Ground reference for all signals |

6.1.6 RJ12 Serial Connector

The serial connector is a 6 pin RJ12 connector as show below on the front panel of the XciteE driver and is labelled X4.

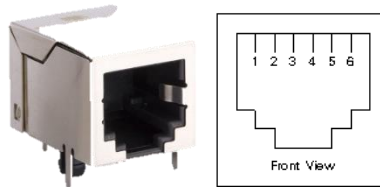


Figure 6.1.6-1 Serial Connector (X4)

| Pin # | Pin Type | Pin Name | Description |
|-------|-----------|----------|---------------------|
| 1 | Output | DBGTX | Debug UART Transmit |
| 2 | Reference | GND | Signal Ground |
| 3 | Output | TX | Main UART Transmit |
| 4 | Input | RX | Main UART Receive |
| 5 | Reference | GND | Signal Ground |
| 6 | Input | DBGRX | Debug UART Receive |



6.1.7 RJ45 Field IO Connector

The Field IO connector is an 8 pin RJ45 connector as show below on the rear panel of the XcitE driver and is labelled X2. The port was designed to be an expansion port that could support multiple protocols. This connector is on all models, but will function as an Ethernet I/P interface if the device serial number ends with “-E”. Units without this suffix will not support any communication through this port.

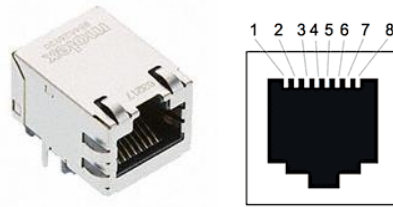


Figure 6.1.7-1 Field IO Connector (X2)

Ethernet/IP Bus Pin Out

| Pin # | Pin Type | Pin Name | Description |
|-------|----------|----------|----------------|
| 1 | IO | TX+_D1 | Transmit Data+ |
| 2 | IO | TX-_D1 | Transmit Data- |
| 3 | IO | RX+_D2 | Receive Data+ |
| 4 | | Reserved | Future |
| 5 | | Reserved | Future |
| 6 | IO | RX-_D2 | Receive Data- |
| 7 | | Reserved | Future |
| 8 | | Reserved | Future |



7 Troubleshooting

The LEDs on the front panel can tell you a lot about what's going on inside the XciteE Driver.

If the ERR light (red) is **on** or **flashing**, please refer to the **Diagnostic Error Codes** section.

If the ERR light (red) is **off**, the XciteE Driver has not detected an error – the following cases apply:

| Light Status | Description |
|--|--|
| No RDY Light | Check Power Connections (Bad Power Connection?) |
| No TX or RX during serial comm - Not responding to serial messages | Check PC software config (Not transmitting to XciteE Driver?) Check serial cable (Transmit not connected to XciteE Driver?) |
| TX light flickers on serial comm, no response from PC | Check message protocol (XciteE Driver will not respond to invalid message formats or data rates) |
| TX/RX lights flicker on serial comm, no response from PC | Check PC software config (Not receiving from XciteE Driver?) Check serial cable (Receive not connected?) |
| Sync Light not flashing | Check if RPM variable > 100, SYNC_MODE = 1 Check signal at WINPUTA pin |
| Sync Light flashing, Firing light not flashing | Check if waveform table has valid data |
| Injector not making noise | Check FIRING_ANGLE, OFFSET_ANGLE, BOOST_VOLTAGE |
| Injector sound, response time, or discharge erratic | Check current waveform for stability Check external I/O connections, grounding Check value of ZENER_VOLTAGE – usually a higher value is better – on the order of BOOST_VOLTAGE |

Troubleshooting Chart



8 Warranty

The XciteE Driver is warranted against any craftsmanship flaws or software bugs for a one year period commencing the day of sale. Warranty does not cover items that have become damaged due to customer installation, placement, connection to utilities or other activities prior to power-up. Defective purchased hardware is guaranteed for the duration of the manufacturer's warranty. Items replaced under warranty need to be submitted with an RMA number to MCM for inspection prior to receiving credit. Items damaged due to abuse or mishandling are excluded from warranty coverage. Software updates that exceed the scope of the original specifications are not considered warranty items. Software fixes that are under warranty will merit updated documentation.

9 Appendix A – Physical Data

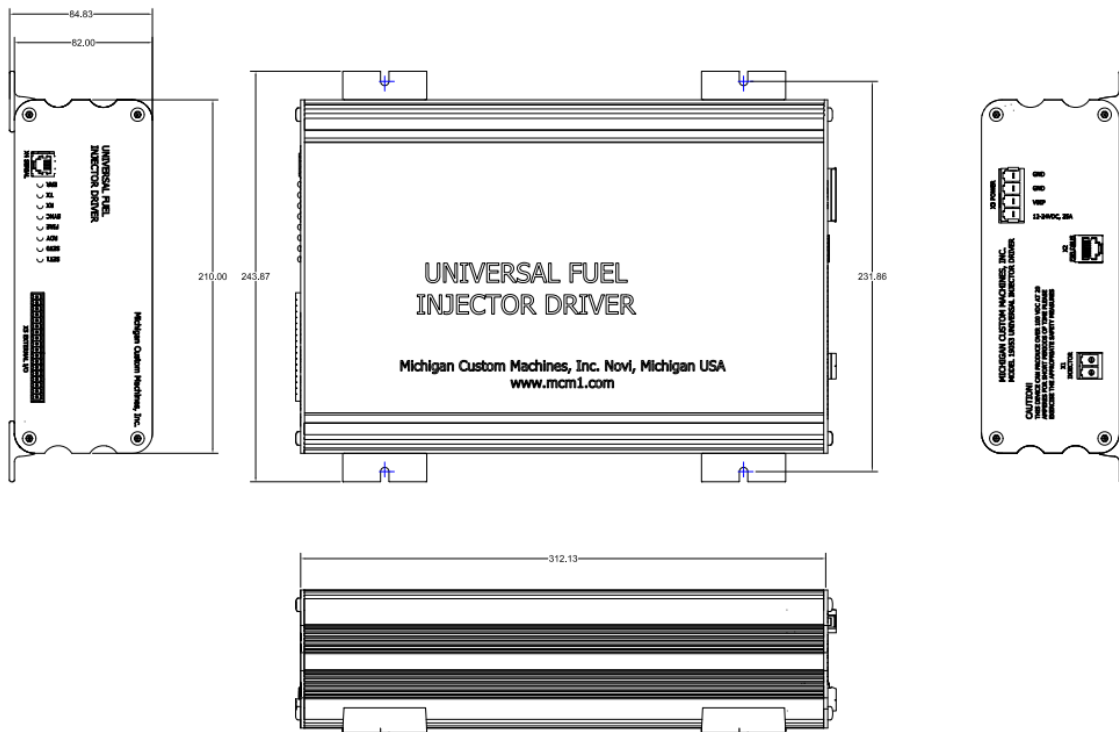


Figure 6.1.7-1 XciteE Driver Physical Dimensions



10 Appendix B – Specifications

Specifications

| | |
|-----------------------------------|--|
| Utilities | |
| Electrical Power | 12-24 VDC, 5A-25A |
| Specifications | |
| Maximum number of injectors | 1 |
| Maximum injector current | 30A |
| Maximum boost voltage to injector | 110V |
| Communication | RS232 and Ethernet IP |
| Serial protocol | Structured, loosely based on SAE J1708 protocol |
| Maximum speed | 6000 RPM (largely depends on duration of injection) |
| Environment temperature | 10 – 45 C (50 – 113 F) |
| Waveform modes | Standard or BIP |
| Sync modes | Asynchronous, Synchronous to OPR or patterned wheel |
| Waveform Control | |
| Minimum time between phases | 1µs |
| Maximum number of phases | 20 |
| Chop Amplitude range | 0-2A |
| Current range | 0-30A |
| Phase duration range | 0-65ms |
| Zener voltage range | 0-110V or disabled |
| Zener control | All drops or only when dropping to zero |
| Boost voltage control per phase | 1) Use programmed boost voltage 2) force off 3) force on |
| Connections | |
| Serial port | RJ11 (DB9 to RJ11 cable and software included) |
| Ethernet IP | RJ45 |
| Power | Screw terminals on removable connector |
| Injector | Screw terminals on removable connector |
| Wave form selection | Screw terminals on removable I/O connector |
| Hall effect switch, OPR | Screw terminals on removable I/O connector |
| Mechanical | |
| Dimensions | 210 x 313 x 82mm (8.25 x 12.5 x 3.25") |
| Mounting | 4 adjustable mounting feet |
| Weight | 1.36 kg (3 lbs.) |
| Construction | Powder coated aluminum housing |

Figure 6.1.7-1 Specification Table

11 Appendix C Front and Rear Labels

The following figures identify front and rear labels on the XcitE driver

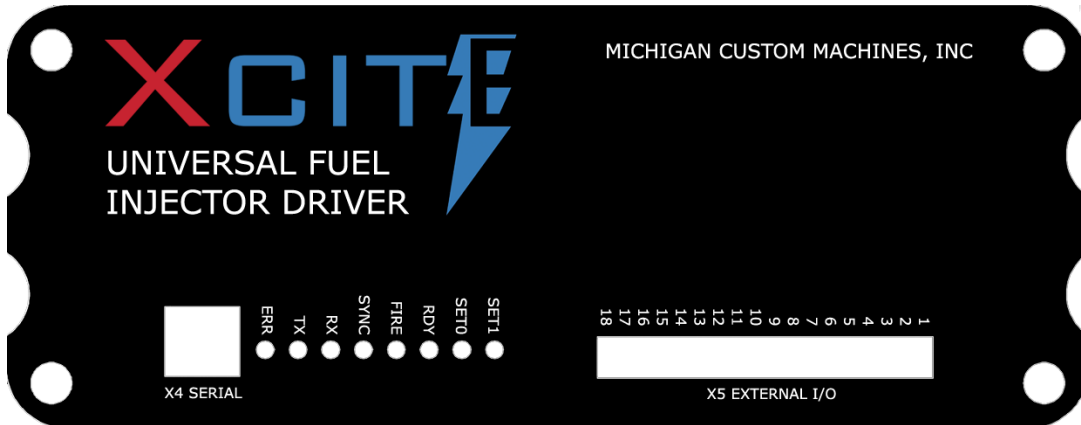


Figure 6.1.7-1 XciteE Driver Front Label



Figure 6.1.7-2 XciteE Driver Rear Label