



I²C SUB-ADDRESSING TRIGGER MADE EASY (LAB_WRX110)

The triggering capabilities included as part of the LeCroy I2Cbus TD package allow you to quickly isolate address and data patterns present in the bus traffic within your design. In the case of sub-addressing schemes, this capability is very useful in identifying particular address spaces used internally by the device.

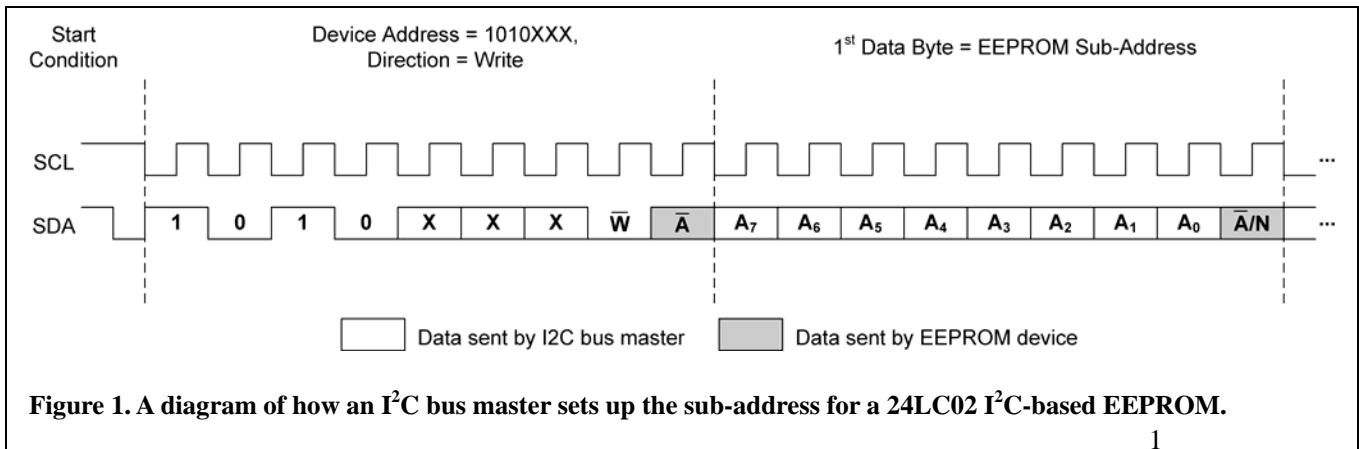
“Sub-addressing” is a term used to refer to the notion that, often, I²C-based devices will utilize internal addresses that are independent from the device address, as defined in the I2C specification. In order to support this internal addressing scheme, the device manufacturer defines a communications protocol that uses bytes within the data pattern to define these addresses. This is especially common within devices such as EEPROMs, which rely very heavily on sub-addressing.

An example of this is the 24LC02 I²C EEPROM. This device uses a single data byte (the first data byte sent during write frames) in order to set an internal address pointer for subsequent memory accesses (see Figure 1). By constructing trigger

conditions that focus on this data byte, you can use the I2Cbus TD package to identify when certain parts of the memory are being accessed, allowing you to gain greater visibility into your memory subsystem.

These triggers can be set up very easily using the I2Cbus TD package. Within the I²C trigger menu, either the ADDR+DATA or EEPROM mode triggers can be used to set these conditions. The major difference between the ADDR+DATA and EEPROM mode triggers is that the EEPROM mode is geared towards searching frames with large amounts of data for specific patterns. An important added benefit of the EEPROM mode trigger is the ability to constrain the trigger data pattern to start at a particular byte. This is useful for our purposes because we want to check only the first data byte for the sub-address, and we can do this by setting the Start Byte to 0.

An example in Figure 2 on the next page, is shown using the EEPROM trigger mode.



The ADDR is set to “1010xxx”, which is the device address used by the 24LC02. The direction is set to write. This should be the case whether you are searching for EEPROM reads or writes, as the sub-address is always written by the bus master to the EEPROM device. In this example, the DATA pattern is set to 0x20, which means that any write or read to EEPROM sub-address 0x20 will result in the trigger condition being met. Using the I2Cbus TD decode function, you can quickly verify the values being written or read from this address in either binary or hexadecimal format.

In some cases, it may be more useful to trigger on a subset of memory addresses and look for reads or writes to any address within the block. In cases where the block is aligned to upper address bits, a simple DATA = trigger with bitmasks can be used to identify the block. For instance, in order to trigger on a sub-address between 0x48 and 0x4F, use a DATA pattern of “01001xxx”.

However, in cases where the block does not align to upper address bits exactly, the conditional data triggering options can help you identify the block in question. For instance, to trigger on a sub-address between 0x16 and 0x1D, use an INRANGE trigger with 0x16 and 0x1D as your lower and upper bounds, as shown in Figure 3.

By utilizing the simple-to-use yet powerful data triggering options included with I2Cbus TD, you are able to identify the sub-addresses you care about quickly.

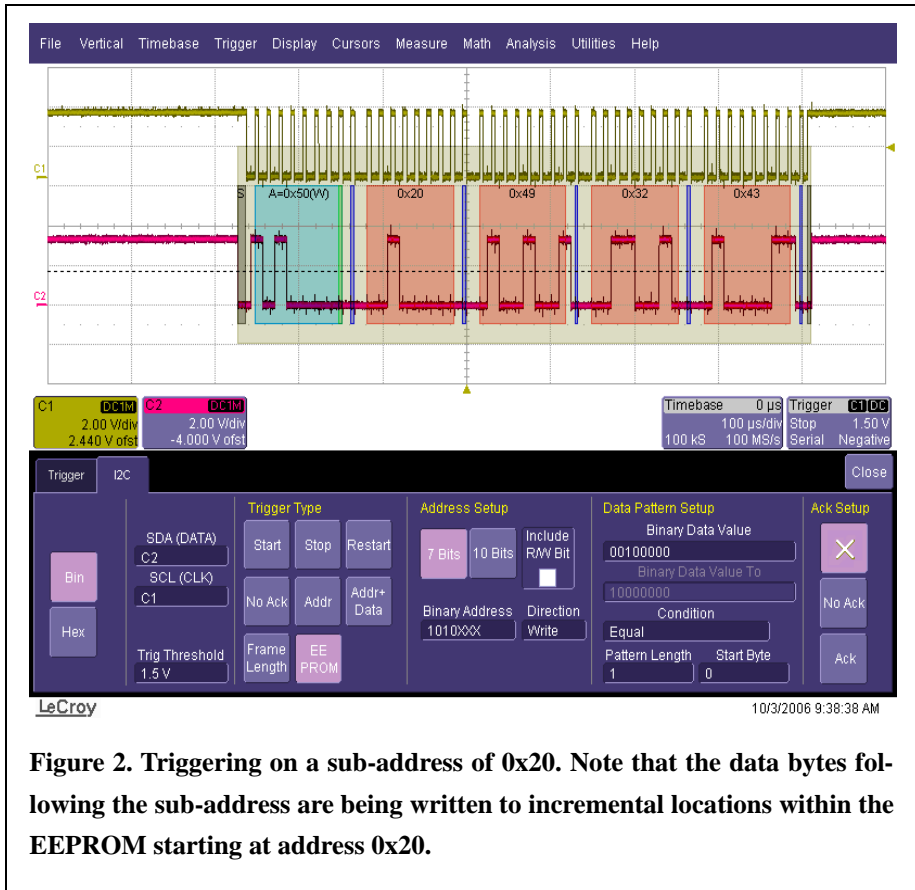


Figure 2. Triggering on a sub-address of 0x20. Note that the data bytes following the sub-address are being written to incremental locations within the EEPROM starting at address 0x20.

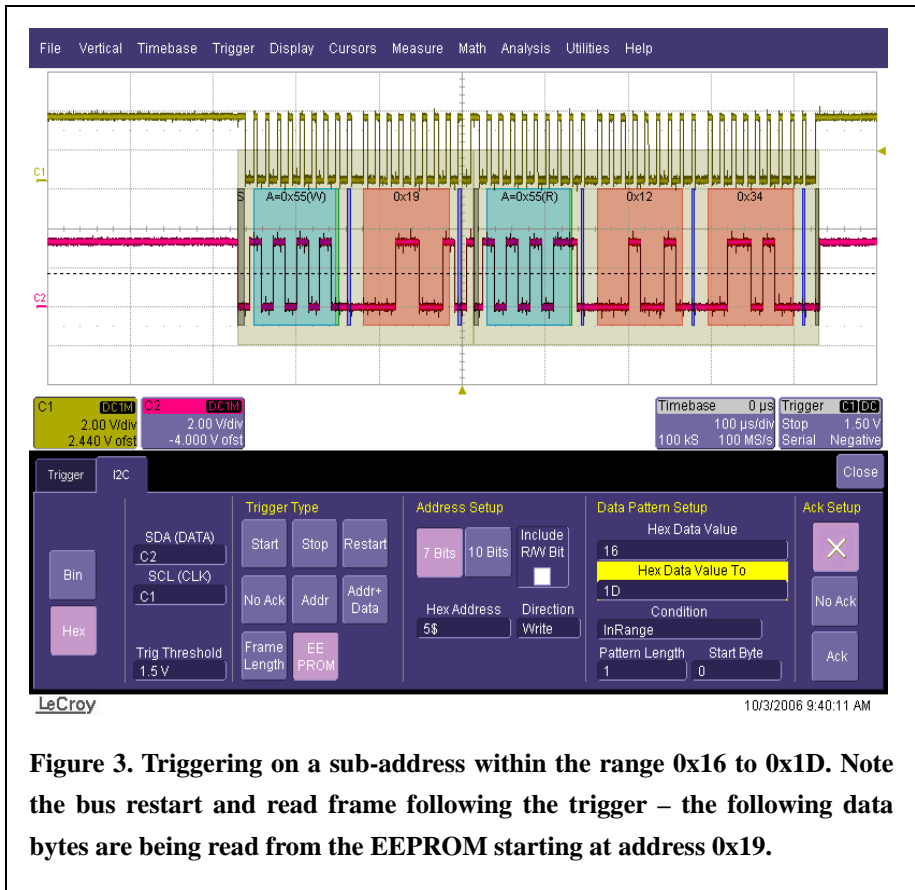


Figure 3. Triggering on a sub-address within the range 0x16 to 0x1D. Note the bus restart and read frame following the trigger – the following data bytes are being read from the EEPROM starting at address 0x19.