I2C Protocol Interface for ARM Microcontrollers

Industrial Sorter Project

Cody Birkland, EWU IEEE Student Chapter

August 16, 2020

Problem:

We need to interface with the color sensor for the first-stage development of the Sorter Project. The TCS34725 Color Sensor uses the I2C protocol for control signals and data, and the TM4C has multiple integrated I2C controllers. We need to understand the details of the protocol such as the signal characteristics, symbol meanings, and other information specific to the color sensor and the microcontroller.

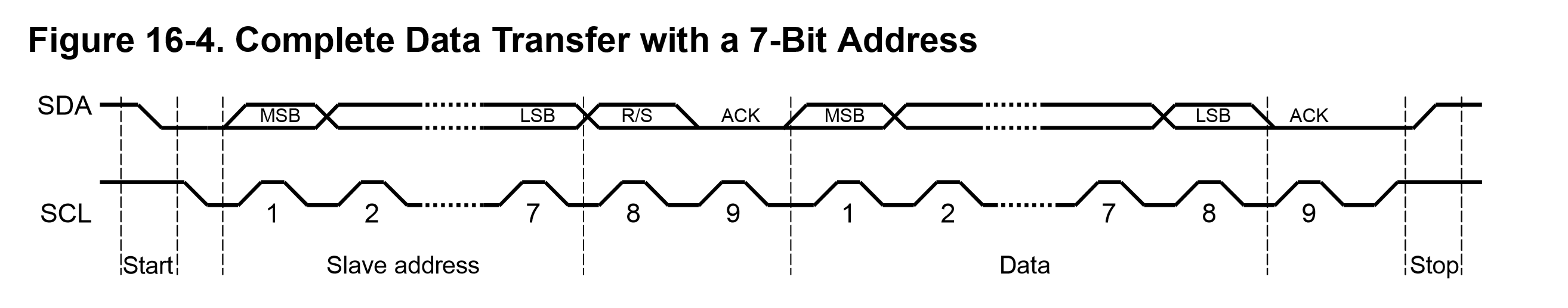
Research:

The details of the I2C Protocol were researched online. The reference used are:

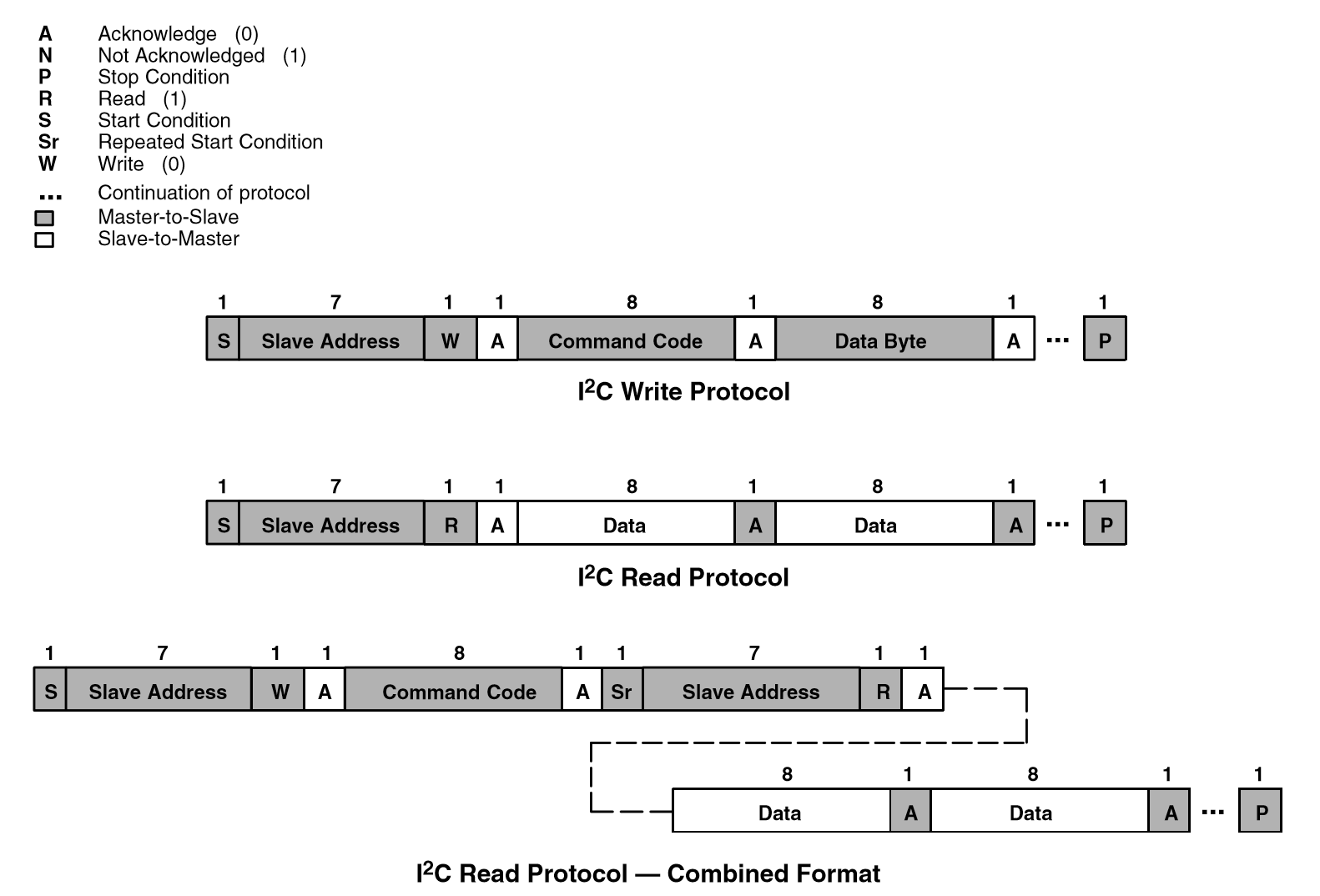
* I2C Basics: <https://www.allaboutcircuits.com/technical-articles/introduction-to-the-i2c-bus/>
* I2C DigiKey article: <https://www.digikey.com/en/articles/why-the-inter-integrated-circuit-bus-makes-connecting-ics-so-easy>
* More reading and address list: <https://learn.adafruit.com/i2c-addresses>
* TCS34725 Datasheet: <https://github.com/EWU-IEEE-Spokane/Sorter-Project/blob/master/I2C%20Color%20Sensor/TCS34725%20Datasheet.pdf>
* Reference to TM4C Datasheet <https://github.com/EWU-IEEE-Spokane/TM4C-Resources/blob/master/TM4C%20Datasheet.pdf> - pages:
  + - Introduction – Page 997
    - Block diagrams – Page 1008
    - Initialization Procedure – Page 1015
* Info on SCCB (I2C for cameras): <http://forums.parallax.com/uploads/attachments/49380/88241.pdf>

The protocol is summarized here:

* I2C utilizes a simple 2-wire communication channel consisting of a Serial Clock wire (SCL) and a Serial Data wire (SDA). Both signals are connected to the positive supply voltage via pull up resistors.
* Communications occur between a Master Device which controls the bus and one or more Slave Devices that can send and receive data when commanded to. There are provisions for having Multiple Masters in one bus, but this is not common in basic applications.
* Data is sent one byte (8 bits) at a time in a Transmission Session. A transmission starts with a start signal, then is immediately followed by the 7-bit Address of the Slave Device to be communicated with. This is then followed by a single bit that specifies Read or Write. This determines if the next 8 bits are data Written to the Slave Device, or 8 bits of data Read from the Slave Device. Following the data, the receiving device asserts a single bit Acknowledge (ACK) signal that tells the transmitting device that the data was successfully received. With a valid ACK, the transmitting device can send as many successive bytes of data as necessary, each with their own ACK bit that must be valid. Once the data is sent, the Master Device asserts a Stop signal that terminates the Transmission Session.



* Communications can occur in Two modes: A Single Start mode, where an initial Start, Address, and Read/Write command is followed by one or more bytes of data. In this mode, all data in the Transmission is either Read from or Written to the Slave Device. There is also a Repeated Start mode, where multiple Start signals can be used to switch between Read and Write modes without stopping the Transmission.



* There are 4 legal symbols: Start, Data High, Data Low, Stop
* START: Clock high, Data transitions from high to low
* STOP: Clock high, Data transitions from low to high
* Data is either low or high. It is read whenever the clock is high. The data is only permitted to change when the clock is low.
* The ACK signal is the bit immediately following the Data byte, and is low for a successful Acknowledge. This bit behaves the same as a Data bit.

Action:

After learning the intricacies of the protocol, the TM4C Datasheet was used as a guide to develop C code to initialize an I2C module on the Microcontroller then establish a working communication channel with the TCS34725 Color Sensor. After significant debugging, the code was developed to enable reliable communications with the sensor in Single Start mode only.

Initial Code: <https://github.com/EWU-IEEE-Spokane/TM4C-I2C>

For reasons that have yet to be fully understood, the code implementing a Multiple Start mode driver was not operational and would fail to change the mode of operation. The code also lacks failure handling and an intelligent bus monitoring function. For now, there are significant delays to allow ample time for the communications to complete before moving on. This will need to be optimized in the future.

Value:

Having a working interface for the Color Sensor allows us to move forward with the development of the sensor aspect of the project. This is a vital part of the sorter machine and getting the sensor code working is a major milestone. Additionally, developing an understanding of the I2C protocol will be useful for interfacing with the camera as development continues, since the SCCB protocol that it uses is closely related to I2C.

Beyond this project, the framework for a general-purpose I2C driver for the TM4C will give us a major start towards working with a multitude of different devices. I2C is such a popular communication protocol that there are thousands of devices that use it out there, covering almost everything imaginable such as external memory, accelerometers, and display devices. Before pursuing the many applications out there, we should refine the code to be general purpose, fault tolerant, efficient, and supportive of the Multiple Start mode.