

COLOR READER

PRODUCT DESIGN SPECIFICATION

Designers: *J. Ma, T. Ho*

Version: *0.2*

05/02/2020

TABLE OF CONTENTS

1 INTRODUCTION.....	4
1.1 Purpose of The Product.....	4
2 GENERAL OVERVIEW AND DESIGN GUIDELINES/APPROACH	4
2.1 Assumptions / Constraints / Standards	4
3 ARCHITECTURE DESIGN.....	4
3.1 Logical View	5
3.2 Hardware Architecture	5
3.3 Software Architecture.....	6
3.4 Communication Architecture.....	6
3.5 Performance Considerations	7
3.6 Power considerations.....	7
3.7 SENSORS/aCTUATORS DESCRIPTION.....	8
4 SYSTEM DESIGN.....	8
4.1 Use-Cases	8
4.2 Data Conversions	8
4.3 Application Program Interfaces	8
4.4 User Interface Design	8
4.5 Performance	9
4.6 Bill of material (BOM)	9
4.7 Calibration and test procedures	9
5 CONCLUSION ON PRODUCT DESIGN SPECIFICATION.....	10
APPENDIX A: REFERENCES	11
APPENDIX B: KEY TERMS	12

1 INTRODUCTION

1.1 PURPOSE OF THE PRODUCT

This product is an integrated cost-effective multi-spectral sensor-on-module solution that can provide highly accurate and calibrated readings for certain wavelengths so that their optical characteristics can be tuned and further characterized upon customer's needs.

2 GENERAL OVERVIEW AND DESIGN GUIDELINES/APPROACH

This product can generally be used to sense wavelengths from approximately 430nm to approximately 670nm (i.e. visible spectrum). There are six channels that can be read: Violet at 450 nm, Blue at 500 nm, Green at 550 nm, Yellow at 570 nm, Orange at 600nm, and Red at 650nm. The source of data can either be a direct illumination source (i.e. ideally 5700K white LED) or the light intensity that is reflected off the surface of any colored object.

There are two different modes of reading for each wavelength, namely raw mode and calibrated mode. In either mode, a) each read data comes from several memory-mapped (or virtual) registers whose bytes can be concatenated together, b) these registers are grouped together in the Big-Endian format (i.e. the most significant byte located at the least significant address), c) the minimum integration time for each spectral conversion is 2.8msec (i.e. fastest mode, least precise), and d) the maximum integration time for each spectral conversion is 714msec (i.e. slowest mode, most precise). In the raw mode, there are two 8-bit registers that together represent an integer value for each wavelength reading. In the calibrated mode, there are four 8-bit registers that together represent a floating-point (i.e. more precise) value for each wavelength reading.

2.1 ASSUMPTIONS / CONSTRAINTS / STANDARDS

The general use cases of the current release of this product can be best guaranteed if the user will a) provide a robust power supply (5V) that can meet all of the minimum current-consuming requirements of the Atmega324PA, of the AS7262, and of the TFT LCD, b) operate this product in the appropriate range of operating conditions (i.e. temperature) that are allowed for each component, and c) minimize exposure to high-energy UV environments (i.e. upward looking outdoor applications). Another assumption is that when the user takes a reading either in raw mode or calibrated mode, the source of the spectral data should stay relatively constant from the time the command is issued to the time the spectral conversion process is complete. This is critical for accuracy because care should be taken to make sure that the integration values from both banks are derived from the same spectral conversion cycle, which may take a little more time because spectral data from different conversion cycles will not be reported.

3 ARCHITECTURE DESIGN

This section outlines different architectural views of the design for the color reader. In particular, a flow chart describes a high-level flow of operations, a block diagram describes interactions between components, and a UML-like diagram shows abstract implementations of the inner workings of the system.

3.1 LOGICAL VIEW

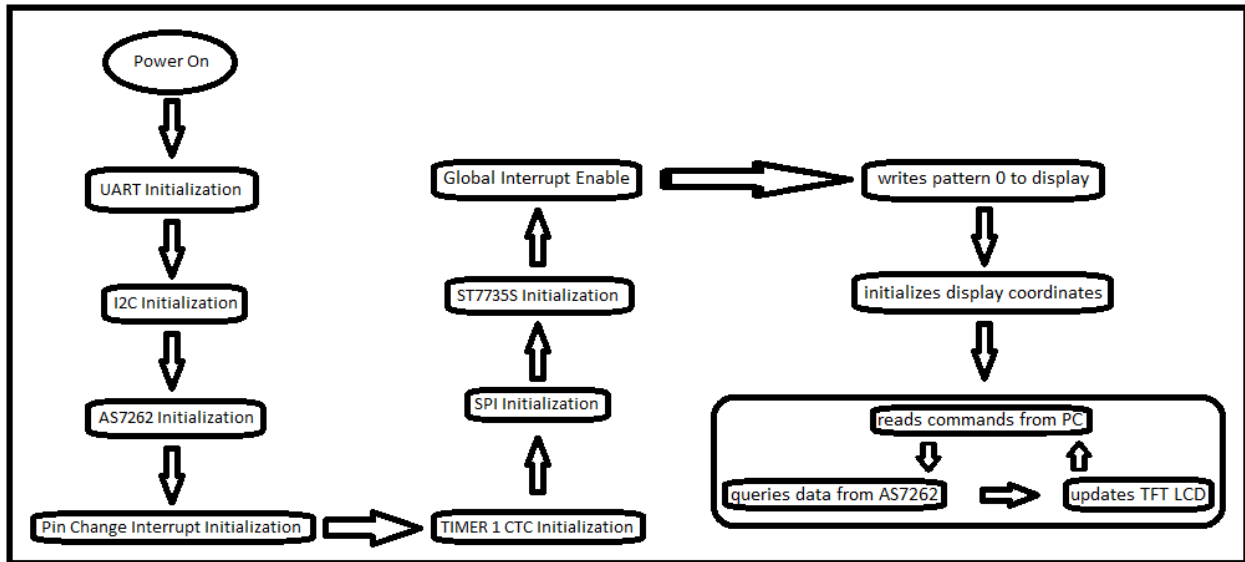


Figure 1: A logical view for the high-level operation of the color reader.

The system starts its operation (Figure 1) after power-on or hard reset. After a series of initializations are carried out (for USART, I2C, AS7262, Pin Change Interrupt, TIMER1 CTC, SPI, and ST7735S), the system is ready to query spectral data from the AS7262 break-out module and to record the results on the TFT LCD module.

3.2 HARDWARE ARCHITECTURE

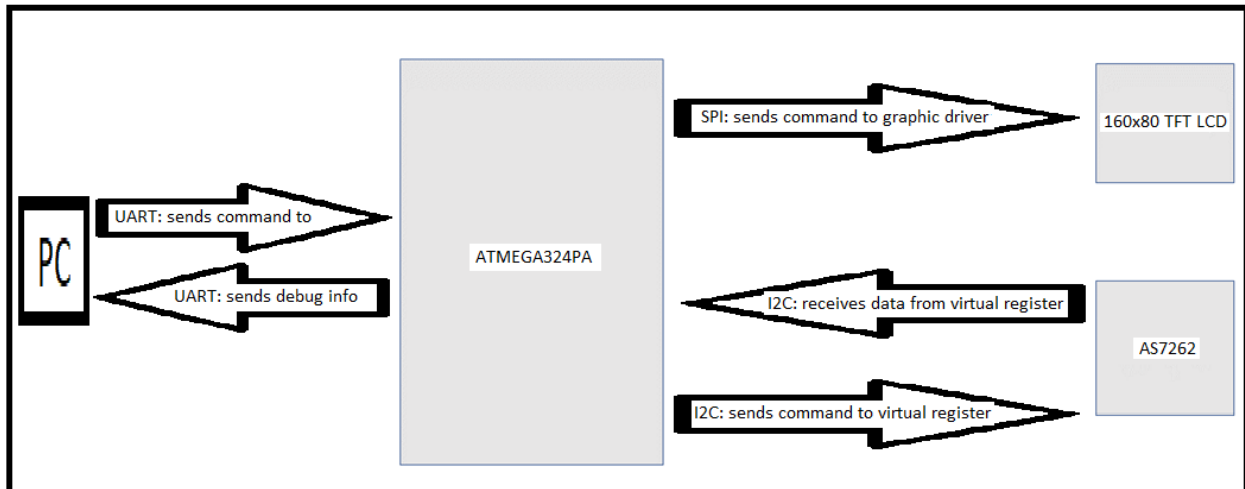


Figure 2: A block diagram for the high-level hardware architecture of the color reader.

The hardware architecture (Figure 2) consists of an Atmega324PA μ C, an AS7262 module, a 160x80 TFT LCD, and a PC. The μ C receives commands from the PC to send specific queries to the AS7262 module. The μ C receives data from the AS7262 module and sends data both to PC for debug and to TFT LCD for display.

3.3 SOFTWARE ARCHITECTURE

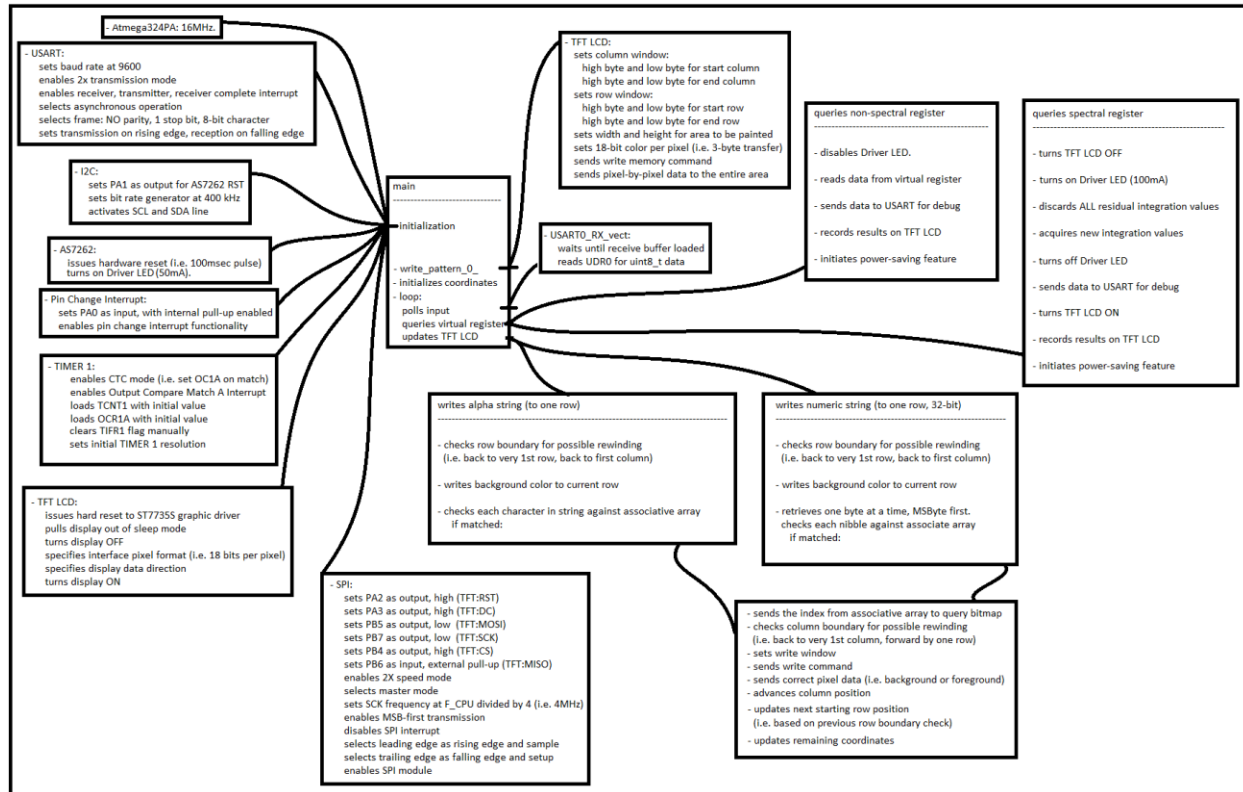


Figure 3: A UML-like diagram for the high-level software architecture of the color reader.

The main function first initializes individual modules, writes a background pattern to the TFT LCD, and initializes coordinate values before any pixel can be written to the TFT LCD. An infinite loop is entered so that the μC can read commands from the PC via USART protocol, query virtual registers of the AS7262 module via I2C protocol, and write text to TFT LCD via SPI protocol. Abstract implementations are presented in Figure 3 and specific implementations are included in the source code.

3.4 COMMUNICATION ARCHITECTURE

The μC communicates with the PC via USART protocol (Figure 2). In particular, the USART module operates at a baud rate of 9600 with the 2x transmission mode enabled. Each transmitted frame consists of one start bit, 8 bits of data, and one stop bit. A receiver complete interrupt is used so that the μC can receive one ASCII character at a time from a PC. A transmitter is also enabled so that the μC can send debug data to the PC.

The μC communicates with the AS7262 module via I2C protocol (Figure 2). In particular, the SCL frequency is configured at 400 kHz, and both clock line and data line are pulled up externally. For read, the μC has to poll the STATUS register of the AS7262 for the TX_VALID bit, which indicates when a query can be sent to the interface. The μC then sends a query to the target virtual register by writing the register's address to the WRITE register of the AS7262 (i.e. when the interface is ready) to issue a pending read request to that register. The μC then polls the

STATUS register of the AS7262 for the *RX_VALID* bit, which indicates when the result of a query is ready. The μ C can then read the queried data from the *READ* register of the AS7262 once this register has been populated with result of a query. For write, the μ C must poll the *STATUS* register of the AS7262 for the *TX_VALID* bit, which indicates when a query can be sent to the interface. The μ C then sends a query to the target virtual register by writing the register's address to the *WRITE* register of the AS7262 (i.e. when the interface is ready). However, the most significant bit of the register's address must be set to a high level to issue a pending write request to that register. The μ C then polls the *STATUS* register of the AS7262 for the *TX_VALID* bit, which indicates when the input data can be written to the target virtual register. The μ C can then write data to the *WRITE* register of the AS7262 to load data into the target virtual register.

The μ C communicates with the TFT LCD via SPI protocol (Figure 2). In particular, the SCK frequency is configured at 4 MHz, the MSB is transmitted first at 2x speed mode, the leading edge is the rising edge for sampling, the trailing edge is the falling edge for setup, the μ C is the master and the TFT LCD is the write-only slave. To write a character at a given resolution to the TFT LCD, the μ C first needs to set a window to be painted and send the bounding coordinates to the TFT LCD. The command is sent by setting the D/C line to a low level, pulling the CS line low, loading the SPDR register with the command opcode, waiting for the transmission to complete (i.e. by checking the SPIF bit in SPSR register), and terminating each transaction by pulling the CS line high. Each parameter for any given command can be sent next by setting the D/C line to a high level, pulling the CS line low, loading the SPDR register with the command opcode, waiting for the transmission to complete (i.e. by checking the SPIF bit in SPSR register), and terminating each transaction by pulling the CS line high. After the paint window has been configured, the write memory command can be sent followed by multiple 2-byte or 3-byte pixel transfers to the TFT LCD.

3.5 PERFORMANCE CONSIDERATIONS

There are at least two factors that can limit the performance of this device. One is the amount of current consumption by the Driver LED. For example, it has been observed that if the LED is configured at 12.5mA then the spectral data is not as accurate as when the LED is configured at 100mA. Another potential limiting factor for the accuracy of the spectral data is the value of the integration time. For example, it has been observed that the spectral data are more accurate with larger integration time (i.e. max at 714msec) and less accurate with smaller integration time (i.e. min at 2.8msec).

3.6 POWER CONSIDERATIONS

There is at least one power constraint on this device. In particular, the amount of current consumption of the Driver LED (i.e. helps determine spectral conversion accuracy) can severely limit how much current can be drawn by other components before the system becomes unstable. For example, when the Driver LED from AS7262 is configured at 100mA and the TFT LCD is also turned on, it has been observed that the system would become unstable due to voltage droop. However, two work-around solutions have been found to address this limitation.

One solution is to limit the maximum current consumption of the Driver LED to only 50mA, but this would reduce the accuracy of the spectral conversion process. Another solution is to enable either the Driver LED or the TFT LCD screen at any given time (i.e. not both activated at the same time), but this would reduce the interactive-ness of the device.

Apart from the limitation of having a potentially weak power supply, a power-saving feature has been implemented so that this portable device can last a little longer when operated from a compatible battery. In particular, the Driver LED will be turned off a) when non-spectral data are queried, b) after spectral data are queried, and c) after a certain amount of time of inactivity has elapsed (i.e. 10 seconds). The TFT LCD screen is only turned on to display results of a recent query and then turned off after a certain amount of time of inactivity has elapsed.

3.7 SENSORS/ACTUATORS DESCRIPTION

The USART interface of the μC is the main sensor that is used in the current release of this product to drive the dynamics of the system. This is how the user input can be propagated from the keyboard of a PC, thru a serial interface, and into the μC . From there, the logics of the software will determine which query should be issued to the AS7262 and what string should be written to the TFT LCD.

4 SYSTEM DESIGN

4.1 USE-CASES

There are several use cases for this device such as a) portable spectrometry where any sample material can be categorized based upon the spectroscopic profiles (within certain visible spectrum) of its constituents, b) horticulture where the coloration of vegetation can indicate certain properties of certain natural entities, c) color matching/identification where certain objects can be sorted according to the presence of certain wavelengths that are reflected off their surface, and last but not least d) precision color tuning/calibration where certain aesthetic features are considered top priority for the client.

4.2 DATA CONVERSIONS

The initial spectral conversion process yields calibrated results in a 4-byte format. To translate this encoding into a more human-readable form, software processing is needed to obtain a float value. The actual formulation is provided in the AS7262 datasheet and the implementation is included in the source code.

4.3 APPLICATION PROGRAM INTERFACES

The AVR Dragon Board is the main programmer for the Atmega324PA via JTAG interface. All compile and deploy tools are provided by the Atmel Studio integrated development editor program.

4.4 USER INTERFACE DESIGN


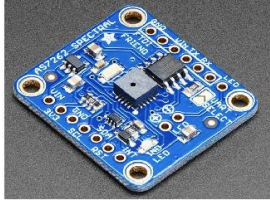

The acquisition of spectral data can be accomplished by issuing individual commands from a PC to an Atmega324PA. There is one ASCII character for each spectral conversion process: 'V' for raw Violet, 'A' for calibrated Violet, 'B' for raw Blue, 'D' for calibrated Blue, 'G' for raw Green, 'E' for calibrated Green, 'Y' for raw Yellow, 'J' for calibrated Yellow, 'O' for raw Orange, 'K' for calibrated Orange, 'R'

for raw Red, 'M' for calibrated Red wavelength, 'N' for all raw spectral data, and 'P' for all calibrated spectral data. There are also non-spectral commands such as: 'H' for hardware version, 'F' for firmware version, 'C' for control register, 'I' for selected integration time value, 'T' for device temperature, 'L' for selected LED control, and 'Q' for screen initialization. The results of each query can then be recorded on a 160x80 TFT LCD.

4.5 PERFORMANCE

A few sample data have been collected from this device and the readings are accurate and precise (i.e. when calibrated data requested). The device performs as intended and will keep integrating data in the background until specific spectral data are requested. The display will update the queried results as soon as data becomes available. Both modes of spectral data (i.e. raw or calibrated) can be queried with ease at any time by simply sending a correct ASCII character from a PC to the μ C.

4.6 BILL OF MATERIAL (BOM)

 <p>ATMEGA324PA-PN 3.38</p>	 <p>AS7262 6-Channel Visible Light / Color Sensor Breakout 19.95</p>	 <p>0.96" 160x80 Color TFT Display w/ MicroSD Card - ST7735 14.95</p>	<p>Total: 38.28</p>
--	--	--	-------------------------

4.7 CALIBRATION AND TEST PROCEDURES

Several colored objects were used as a source for each spectral conversion process. Both raw and calibrated data were compared against each other for consistency. Since calibrated data are possible, there was no need to execute any calibration procedure at this point. The test procedure consisted of detecting the color intensity of various colored objects or sources of illumination on six different channels: violet, blue, green, yellow, orange, and red.

5 CONCLUSION ON PRODUCT DESIGN SPECIFICATION

I have had a great experience with this project because it helps me integrate everything that I have built up from past lab projects. There are a few challenges due to timing issues within each module and between modules (i.e. delayed and mismatched traces). One way to improve the design of this product is to put all the components on a printed circuit board and build an enclosure for all sensors in this color reader.

Appendix A: References

The following table summarizes the documents referenced in this document.

Document Name and Version	Description	Location
megaAVR® Data Sheet	<i>Datasheet for Atmega324PA</i>	http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega164A_PA-324A_PA-644A_PA-1284_P_Data-Sheet-40002070B.pdf
AS7262 v1-01	<i>Datasheet for AS7262</i>	https://www.mouser.com/datasheet/2/588/AS7262_DS000486_2-00-1082195.pdf
ST7735S v1.3	<i>Datasheet for AS7262</i>	https://www.crystallfontz.com/controllers/Sitronix/ST7735S/

Appendix B: Key Terms

The following table provides definitions for terms relevant to this document.

Term	Definition
μ C	microcontroller
<i>LED</i>	<i>Light-emitting diode</i>
<i>TFT LCD</i>	thin-film-transistor liquid-crystal display