

ENSC 305W/440W Grading Rubric for Design Specification

Criteria	Details	Marks
Introduction/Background	Introduces basic purpose of the project.	/05%
Content	Document explains the design specifications with proper justification for the design approach chosen. Includes descriptions of the physics (or chemistry, biology, geology, meteorology, etc.) underlying the choices.	/20%
Technical Correctness	Ideas presented represent valid design specifications that will be met. Specifications are presented using tables, graphs, and figures where possible (rather than over-reliance upon text). Equations and graphs are used to back up/illustrate the science.	/20%
Process Details	Specification distinguishes between design details for present project version and later stages of project (i.e., proof-of-concept, prototype, and production versions). Numbering of design specs matches up with numbering for functional specs.	/15%
Test Plan	Provides a functional test plan for the present project version. (Note that project success will be measured against this test plan.)	/10%
Conclusion/References	Summarizes functionality. Includes references for information from other sources.	/05%
Presentation/Organization	Document looks like a professional specification. Ideas follow in a logical manner.	/05%
Format Issues	Includes letter of transmittal, title page, abstract, table of contents, list of figures and tables, glossary, and references. Pages are numbered, figures and tables are introduced, headings are numbered, etc. References and citations are properly formatted.	/10%
Correctness/Style	Correct spelling, grammar, and punctuation. Style is clear concise, and coherent. Uses passive voice judiciously.	/10%
Comments		



November 17, 2013

Mr. Lucky One
School of Engineering Science
Simon Fraser University
Burnaby, BC V5A 1S6

Re: ENSC 440 Design Specifications for the AccuTag Wireless Electronic Price Tag System

Dear Mr. One,

Attached is an enclosed document from Precision Wireless with technical specifications which outline the full design procedure of the AccuTag system. This document details the modules needed to implement one or more E-Ink display screens with the ability to display prices and product information wirelessly. The purpose of this product is to reduce the use and maintenance of paper price tags currently in practice in the market today and to introduce a more efficient and convenient method of updating prices in large super market stores.

The enclosed document provides a proof of concept guideline for the design requirements needed for the development of the AccuTag system. It provides a complete overview of both the hardware and software procedures necessary to implement a fully functional prototype of our display system. This proof of concept model is set to meet the market standards of a competent cost effective design.

As a team, we would appreciate your time and consideration in reviewing our proposal and fully encourage questions or feedback which may arise in the process. Should you have any concerns, please feel free to contact me at mmehran@sfu.ca or by phone at (778)-859-8070.

Sincerely,

A handwritten signature in cursive script that reads "M. Mehran".

Mahyar Mehran
Chief Executive Officer
Precision Wireless

Enclosure: Design Specifications for the AccuTag Wireless Electronic Price Tag System

Design Specification for Wireless Electronic Price Tag System

AccuTag By Precision Wireless

Executive Summary

This document details the design specifications of the proof of concept prototype of the AccuTag wireless electronic price tag system. This system consists of two main physical components: an in-store transmitter and a set of digital E-Ink LCD price tags. The purpose of this documentation is to provide the reader with a detailed overview of design criteria in terms of product hardware and software mechanisms along with usability and sustainability considerations.

The digital price tags were constructed using an assortment of products such as the Atmega256RFR2 Xplained Pro board along with a 2.7 inch E-Ink display from Pervasive Displays. The E-Ink display was purchased with a bundle from Texas Instrument called TI EPD Booster Extension Kit consisting of the E-Ink display driver and controller. In terms of hardware design for our product, the production prototype consisted of determining the required connections from each of the mentioned boards and designing a whole new Printed Board Circuit layout which was created through the use of a software known as Altium Designer.

These E-Ink price tags will be wirelessly controlled by a central transmitter using Radio-Frequency transmission which will have access to a catalogue of prices that are constantly up to date. This database will be provided by the supermarket which will then be organized and accessible by the main computer connected to the transmitter. Each tag is identified based on its specific aisle, section and item ID which is also stored in a database. This connection is the basis that ensures the efficiency and convenience of our product. The programming team in charge of this task has ensured a software that is easy to use for consumers so that minimal time and effort will be required for initial startup and maintenance.

Any changes to the database for a specific product will result in a signal being transmitted from the central transmitter unit to the designated digital price tag corresponding to that exact item. This signal contains a packet of information consisting of product name, price and misc. product information. Once the signal has been transmitted successfully to the receivers present on the E-Ink price tags, the signal will be further processed by our LCD programming team. The detection of the packet will trigger an interrupt service which will decode the information and produce changes on the E-Ink display through the use of the Serial Peripheral Interface.

Usability has been taken into full consideration through hardware design as the chosen E-Ink display kit was picked primarily for its visibility and also for convenience in terms of size and power consumption. Sustainability was also a key criteria in design as we envisioned a product that would require the least amount of maintenance possible. This was made possible by the selection of this E-Ink display from Pervasive Displays as power consumption and long lasting lifespan had utmost importance in the AccuTag design. Supermarkets can save thousands of dollars by replacing their normal service routines for paper price tags and implementing a more efficient solution with our digital price tags.

The remainder of this document will provide an in depth analysis of our system's software and hardware design specifications. There will also be additional topics covered in the following outline.

TABLE OF CONTENTS

1	Introduction.....	8
1.1	Scope.....	8
1.2	Intended Audience	8
2	System Specification.....	9
3	Hardware Design	9
3.1	AccuTag Transmitter Prototype	9
3.2	AccuTag Display Prototype	15
3.3	AccuTag Display PCB layout considerations	18
3.3.1	Mechanical Specification	18
3.3.2	General Specification	20
4	Software Design	22
4.1	High Level Design.....	22
4.1.1	Database	23
4.1.2	Host Computer	23
4.1.3	Protocol	23
4.1.4	API Set	24
4.1.5	Central Transmitter Unit	25
4.1.6	Price Tag Units.....	26
4.2	E-Ink Display.....	27
4.2.1	SPI Timing Control.....	28
4.2.2	Writing to Memory	30
4.2.3	Powering on the COG Driver.....	30
4.2.4	Initializing the COG Driver.....	31
4.2.5	Writing Data from the Memory to the E-Ink (EPD) Screen	32
4.2.6	Powering off the COG Driver.....	35
5	System Integration / Test Plan	36
5.1	Test Plan.....	36
5.1.1	Database Test Plan	36
5.1.2	Transceiver Test Plan	36
5.1.3	Range Testing	37

5.1.4	Packet Error Rate(PER) Testing	38
5.1.5	Sensitivity and Output Power Testing.....	39
5.1.6	Reliability Testing	39
5.1.7	Microcontroller Testing	39
5.1.8	E-ink Display Testing.....	41
5.2	System Integration.....	41
5.2.1	System Testing.....	42
	Sources and References.....	43
	Appendix A. Schematics.....	44

GLOSSARY OF TERMS

Central Transmitter Unit	The central transmitter unit is the transceiver that communicates the information from the database to the price tags.
Host computer	The machine that the central transmitter unit is connected to which sends control signals to it based on price changes detected in the database.
LCD and E-ink display	The terms LCD and E-ink display are used interchangeably. The LCD that we have selected for both our proof of concept design as well as the final product is an e-ink display.
EPD	Also referred to in this documentation as E-Ink. Stands for Electrophoretic Display (or e-Paper Display).
DUT	Device under test
PER	Packet Error Rate
dBm	dBm is an abbreviation for the power ratio in decibels (dB) of the measured power referenced to one milliwatt (mW)
Tag	Tag is a term used to refer to the system with a microcontroller, transceiver and the e-ink display.
API	Application programming interface
PCB	Printed circuit board
Pinout	Shorthand for “pin layout” used in Atmel documents
Tcon	Timing Controller
FPL	Front Plate Laminate
SPI	Serial Peripheral Interface (also known as SI)
COG	Chip on Glass
WN	Wireless Node
RCB	Radio Circuit Board
STB	Sensor Terminal Board
PDI	Pervasive Displays Incorporated
MCU	Microcontroller

List of Figures and Tables

Figure 3.1.1: The Sensor Terminal Board with the mounted Wireless Node (WN) or Radio Circuit Board (RCB), which is connected to JTAGICE for microcontroller (RCB/WN) programming[1]

Figure 3.1.2: The Sensor Terminal Board

Figure 3.1.3: Sensor Terminal Board[1]

Figure 3.1.4: Picture of the RCB/WN and EXT0 and EXT1 expansion headers[2]

Figure 3.1.5: RCB/WN block diagram[2]

Figure 3.1.6: ATmega256RFR2 Pinout[3]

Figure 3.1.7: ATmega256RFR2 Block Diagram[3]

Figure 3.1.8: ATmega256RFR2 Transceiver Block Diagram[3]

Figure 3.1.9: ATmega256RFR2 extended feature application schematic[3]

Figure 3.2.1: AccuTag Display Unit, ATmega256RFR2 Xplained pro board[4]

Figure 3.2.2: AccuTag Display Unit, E-Paper Display (EPD)[5]

Figure 3.2.3: AccuTag Display Unit, E-Paper Display (EPD) block diagram[5]

Figure 3.2.4: AccuTagXplained Pro extension header[6]

Figure 3.3.1: AccuTag Display top-view drawing

Figure 3.3.2: EPD screen dimension details provided by Pervasive Displays Inc[7]

Figure 3.3.3: Wiring layout schematic for component placement

Figure 4.1.1: High Level View of the AccuTag system

Figure 4.1.2: Flowchart for the transmitter unit's program logic

Figure 4.1.3: Flowchart for the price-tag unit's program logic

Figure 4.2.1: Overview of Update Cycle for E-Ink Display

Figure 4.2.2: Overview of Control Signals during Update Cycle of E-Ink Display

Figure 4.2.3: Timing Waveform for sending SPI commands using SCLK, SI, and CS

Figure 4.2.4: Procedure for sending a generic SPI command (ie. SPI (0xI₁I₂, 0xD₁D₂))

Figure 4.2.5: Flow Chart for procedure on powering on the COG Driver

Figure 4.2.6: Flow Chart for procedure on initializing the COG Driver

Figure 4.2.7: Example for writing to the E-Ink display in four stages

Figure 4.2.8: Flow Chart outlining full procedure on how to send data to the COG Driver

Figure 4.2.9: Flow Chart outlining full procedure on how to display data to the E-Ink Display using four stages

Figure 4.2.10:Flow Chart outlining full procedure for powering off the COG Driver

Figure 5.1.1:State machine of the performance analyzers

Figure 5.2.1:Figure illustrating the integrated AccuTag system

Figure A1:AccuTag Transmitter Interface (STB)

Figure A2:AccuTag Display Unit top-level hardware diagram

Figure A3:The AccuTag Display_ATmega256RFR2 connections to EXT1

Figure A4:The AccuTag Display _Dual Input power supply

Figure A5:The AccuTag Display Unit Antenna

FigureA6:AccuTag Display part 1_E paper schematic

Table 3.1.1:Extension connector (EXT0/1) mapping[2]

Table 3.2.1: *Extension connector pin layout table*[5]

Table 3.3.1:*Accutag Display dimension parameters*

Table 3.3.2: *Mechanical requirements*

Table 3.3.3: *System Electrical Requirements*

Table 4.1.1:*Price catalogue's table schema*

Table4.1.2:Host to transmitter message format

Table 4.1.3:*List of basic APIs*

Table A1:Bill of Materials

1 Introduction

The AccuTag System by Precision Wireless is an eco-friendly wireless display solution, which is also a feasible replacement for the traditional paper price tags. The system consists of a number of E-ink display tags that are assembled with the Atmel single-chip solution for RF communication and data processing. The AccuTag central transmitter unit will interface with a computer or a server from which it will obtain updated prices for individual items and broadcast the data to the AccuTag price tags. The design specifications for the AccuTag display units and transmitter unit are described in further detail below. This document shall cover all the technical details for each component and provide an overview of the high level design of the system both for the proof of concept prototype as well as the production model.

1.1 Scope

The scope of this document is to fully outline and describe the design specification, features and considerations of the AccuTag System that must be satisfied by Precision Wireless upon the completion of this project. In this document, Precision Wireless provides the rationale behind the AccuTag design choices and various aspects of the overall hardware and firmware integration. The listed specifications will explicitly describe the functionalities of the preliminary (rat's nest) prototype and will layout the foundation for any possible or necessary modifications that must be made in the next phases of the project before manufacturing. This specification also explains and defines the test plans designed to evaluate various functions and features of the devices and software and shall be referred to during future stages of development and testing.

1.2 Intended Audience

The intended audiences of this document include all Precision Wireless team members and potential stakeholders. Design engineers shall use this specification as guidelines for the prototype design and final product manufacturing. Test engineers shall also refer to this document for conducting test plans and to confirm that all the features are functioning correctly and as specified in Functional and Design Specification Reference Guides. Project managers and company stakeholders shall use this document as the reference guide to monitor the progress of product design, testing and implementation phases.

2 System Specification

The AccuTag system by Precision Wireless will allow retailers and supermarkets to replace a portion or all of the price tags in their store with digital tags that can be updated automatically via a price catalogue. For optimal visibility and minimal power consumption the AccuTag price tags use e-ink technology as well as single-chip ATmega256RFR2 RF Transceiver/microcontroller.

The operator of the system will need to connect their database to the AccuTag transmitter unit. The default host program that comes with the system will monitor their database for changes to prices and update the displays accordingly.

Maintenance will be absolutely minimal; the price-tags will need to be re-equipped with new batteries once they're drained and the transmitter may occasionally need some miscellaneous maintenance. Otherwise, once the system is configured correctly the store operators can essentially forget about having to change prices around the store.

3 Hardware Design

3.1 AccuTag Transmitter Prototype

The AccuTag transmitter unit will be located at the main office of the store. The designated employee with proper credentials will be able to change the information displayed on one or more price tags depending how many need to be changed. Figure 3.1.1 shows the Wireless Node (WN), which is assembled on the Sensor Terminal Board (STB). The STB allows the user to interface with the WN through various modes such as the USB to virtual COM port or the general purpose I/O (GPIO). The WN is also referred to as the Radio Circuit Board (RCB) as it performs all the data processing and transmissions using the ATmega256RFR2 that is assembled onto the board. The RCB or WN microcontroller can be programmed through the JTAG pins that are available through the STB. The RCB can receive or transmit data through 2.4 GHz radio frequency transmissions. The STB shown in figure 3.1.2 also adds further capabilities to this transmitter unit if it is programmed to operate in stand-alone mode. This feature would require an external power source or the use of batteries provided with the WN. The Transmitter board is able to interface with the computer through a serial COM port via USB then send and receive information from the main database.

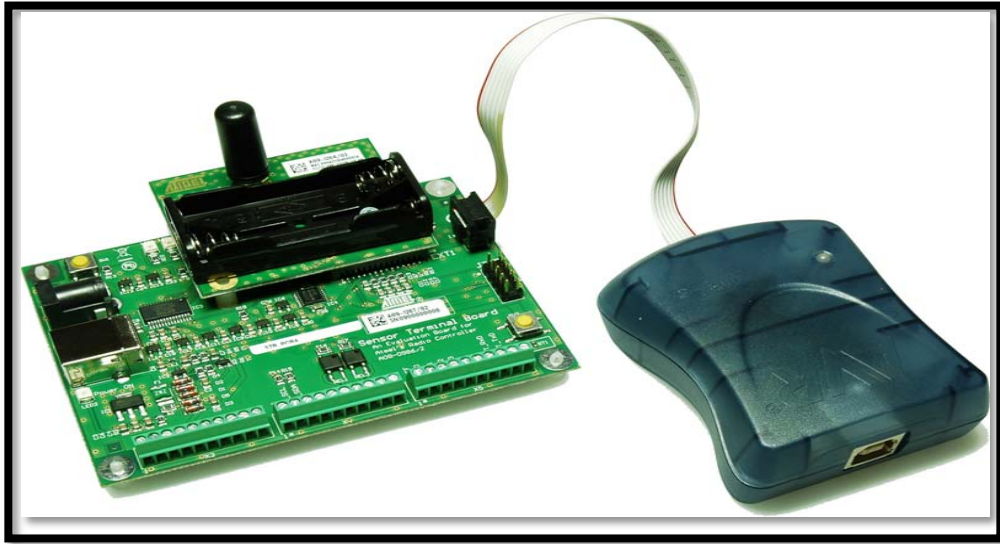


Figure 3.1.1: The Sensor Terminal Board with the mounted Wireless Node (WN) or Radio Circuit Board (RCB), which is connected to JTAGICE for microcontroller (RCB/WN) programming[1]

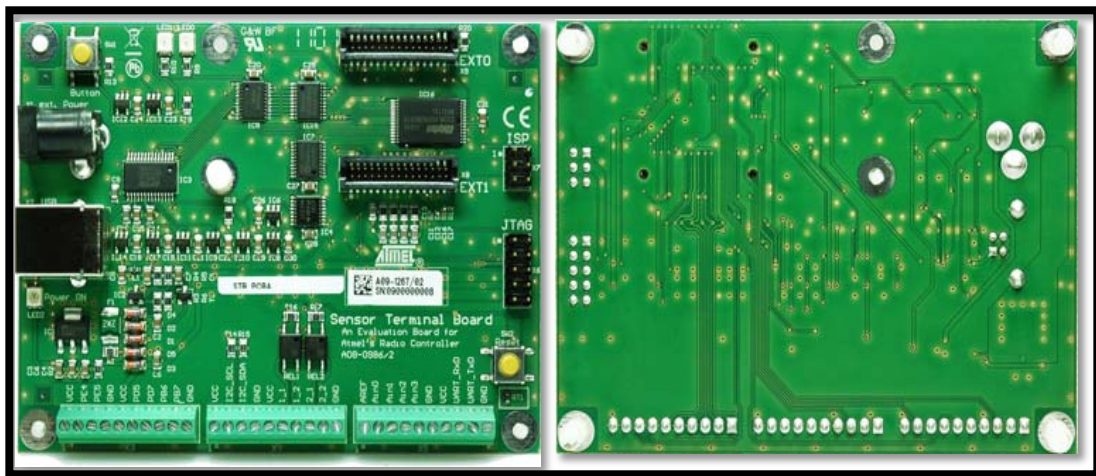


Figure 3.1.2: The Sensor Terminal Board

Figure 3.1.3 below shows various components of the sensor terminal board. The extensions EXT0 and EXT1 are used for interfacing with RCB. After mounting the RCB onto the STB connections (EXT0 and EXT1), the information can be sent or received via the USB to virtual COM port connection. Also ATmega256RFR2 firmware can be programmed/updated through the JTAG port while the RCB is mounted on the STB. The 30-pin EXT0 and EXT1 extensions are also labeled with ATmega256RFR2 pins that they are connected to; this is shown in figure 3.1.4.

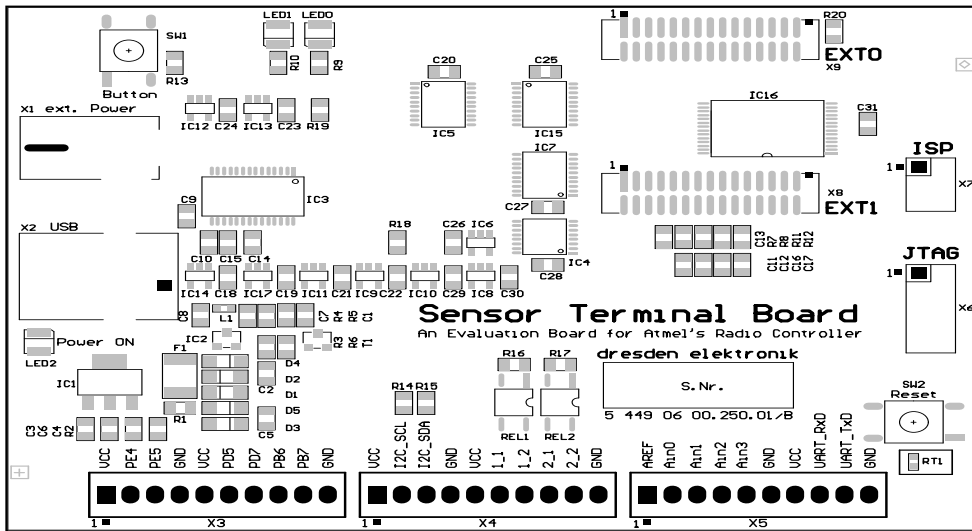


Figure 3.1.3: Sensor Terminal Board[1]

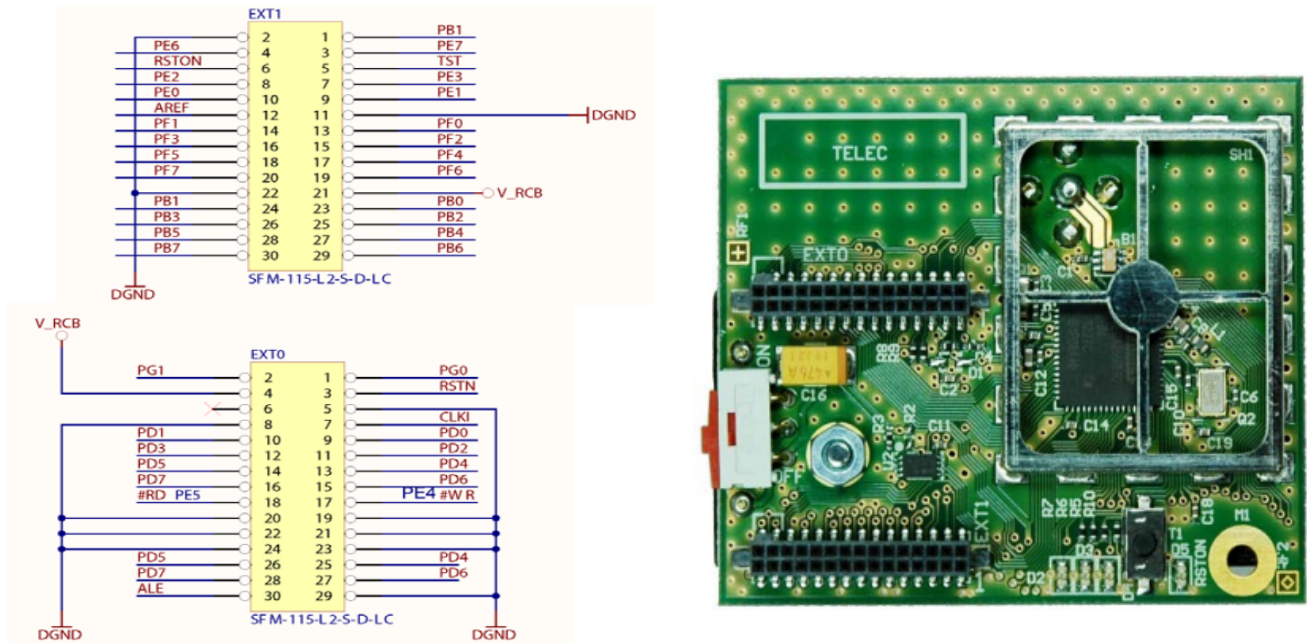


Figure 3.1.4: Picture of the RCB/WN and EXT0 and EXT1 expansion headers[2]

The functions associated with each extension connector pin and the respective Atmega256RFR2 pin number associated with it is tabulated below.

RCB EXT0			
Pin#	ATMEGA256RFR2 pin# and function	Pin#	ATMEGA256RFR2 pin# and function
1	#14_PGO_SPI chip select	2	#15_PG1_DIG1 (Antenna switch-p)
3	#12_RST_reset	4	V_RCB
5	GND	6	Not connected
7	#33_CLKI	8	GND
9	#25_PD0_TWI SCL	10	#26_PD1_TWI SDA
11	#27_PD2	12	#28_PD3
13	#29_PD4	14	#30_PD5
15	#31_PD6	16	#32_PD7
17	#50_PE4	18	#51_PE5
19	GND	20	GND
21	GND	22	GND
23	GND	24	GND
25	#29_PD4	26	#30_PD5
27	#31_PD6	28	#32_PD7
29	GND	30	#16_PG2

EXT1			
Pin#	ATMEGA256RFR2 pin# and function	Pin#	ATMEGA256RFR2 pin# and function
1	#37_PB1	2	GND
3	#53_PE7	4	#52_PE6
5	#11_TST	6	#13_RSTON
7	#49_PE3	8	#48_PE2
9	#47_PE1	10	#46_PE0
11	GND	12	AREF
13	#63_PFO	14	#64_PFI
15	#1_PFI	16	#2_PFI
17	#3_PFI	18	#4_PFI
19	#5_PFI	20	#6_PFI
21	V_RCB	22	GND
23	#36_PBO	24	#37_PBI
25	#38_PBI	26	#39_PBI
27	#40_PBI	28	#41_PBI
29	#42_PBI	30	#43_PBI

Table 3.1.1: Extension connector (EXT0/1) mapping[2]

Figure 3.1.5 below illustrates a block diagram of the RCB, which is composed of the ATmega256RFR2 and some peripheral circuitry. The ATmega256RFR2 RCB uses MAC hardware to handle all the RF modulation and demodulations, signal processing, and frame reception and transmission action [2]. The AccuTag transmitter unit uses the SMA antenna that is available through the RCB.

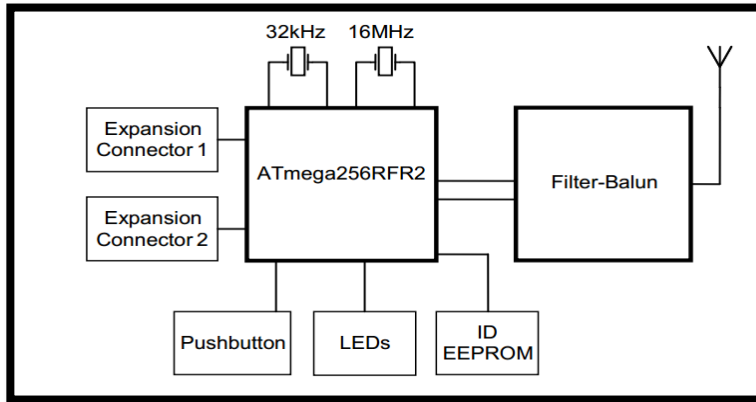


Figure 3.1.5: RCB/WN block diagram[2]

The ATmega256RFR2 is a low-power, single-chip CMOS 8-bit AVR microcontroller, which is combined with a high data rate (up to 2 Mb/s) transceiver for the 2.4 GHz ISM band [2]. The pinout of the ATmega256RFR2 QFN/MLF chip package is shown below. Depending on the application of each AccuTag component (transmitter or display unit) various 8-bit/6-bit bidirectional I/O ports (Ports B, D, E, F and G) can be used. Otherwise the unused GPIO pins must be connected to grounded/power supply through an internal pull-up resistor to prevent power dissipation in the digital input stage and any damage to the chip.

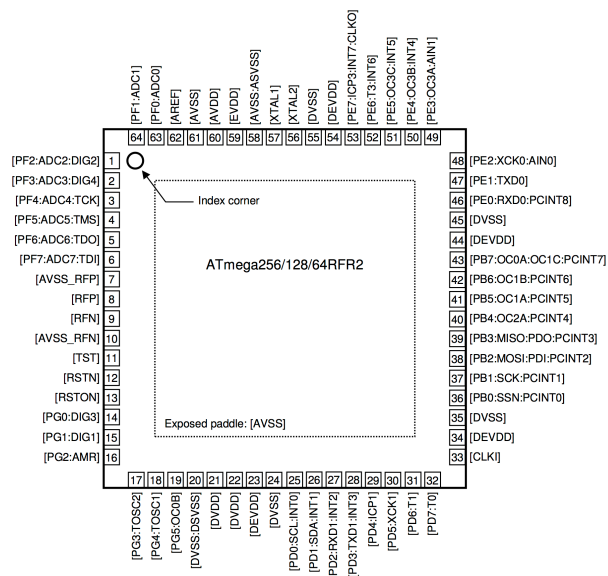


Figure 3.1.6: ATmega256RFR2 Pinout[3]

Figure below illustrates the high-level block diagram of the chip. As mentioned previously the chip is based on AVR enhanced RISC architecture that allows the user to execute powerful instructions in a single clock cycle, hence enabling optimization between power consumption and processing speed. ATmega256RFR2 provides other features including, 256 KBytes of in-system programmable (ISP) Flash with read-while-write capabilities, 8Kbytes of EEPROM, 32Kbytes of SRAM, 35 general

purpose I/O lines, 32 general purpose working registers, 10-bit analog to digital convertor (ADC), and a SPI serial port[3]. The chip uses 16 KHz and 32 KHz crystal oscillators for the system reference clocks.

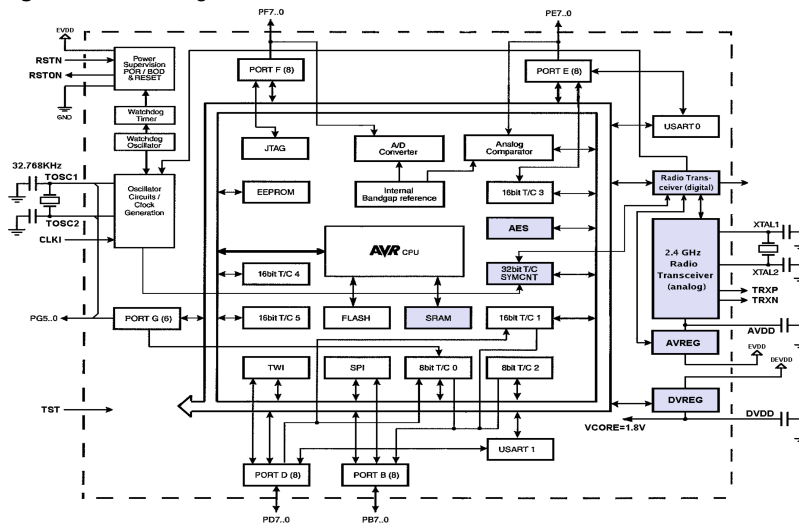


Figure 3.1.7: ATmega256RFR2 Block Diagram[3]

The RF Transceiver uses the 16KHz in-system reference clock and a fully integrated ZigBee solution that allows excellent RF performance with low current consumption. The transceiver allows for complex digital modulation and demodulation including time and frequency synchronization and data buffering. RFP and RFN represent the bidirectional differential antenna pins that are used for data transmission and reception [3]. The figure below illustrates transceiver block diagram.

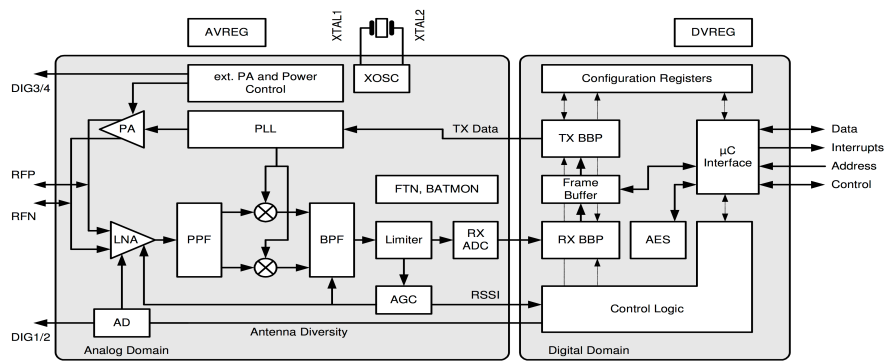


Figure 3.1.8: ATmega256RFR2 Transceiver Block Diagram[3]

The figure below shows an extended feature set application schematic that allows for Antenna Diversity using alternative pin function DIG1/2 at port G and F, Rx/Tx indicator using alternative pin function DIG3/4 at port G and F. Capacitors C1 and C2 provide AC coupling of the RF input to the RF port, and C4 is used to improve matching. CB2 and CB4 are the power supply bypass capacitors that are connected to the external analog supply pin (EVDD, pin #59) and external digital supply pin

(DEVDD, pin #23). Furthermore, capacitors CB1 and CB3 are bypass capacitors used to ensure stable operation and to improve noise immunity of the integrated analog and digital voltage regulators [3]. The AccuTag display unit uses the following schematic and integrates with additional peripheral circuitry (e-paper, LEDs, and push buttons), which is explained in greater detail in the following section of the design specifications.

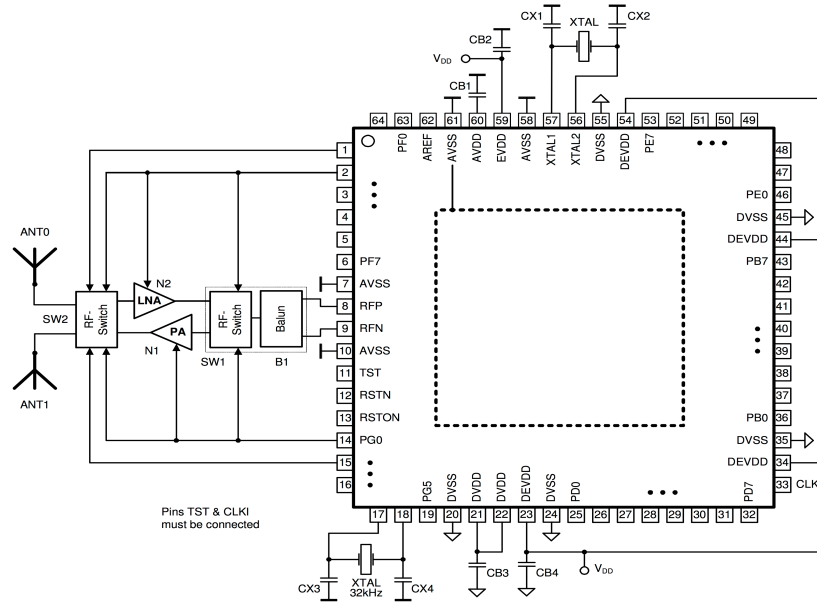


Figure 3.1.9: ATmega256RFR2 extended feature application schematic[3]

3.2 AccuTag Display Prototype

The AccuTag Display unit (rat's-next prototype) uses two components that are integrated together to develop the “rat’s nest” prototype. One of these components is the ATmega256RFR2 Xplained Pro board that uses the ATmega single-chip solution explained above and some peripheral circuitry. This component is responsible for data transmission and processing within this unit. Figure 3.2.1 shows a picture of this board and various components that are available on the board.

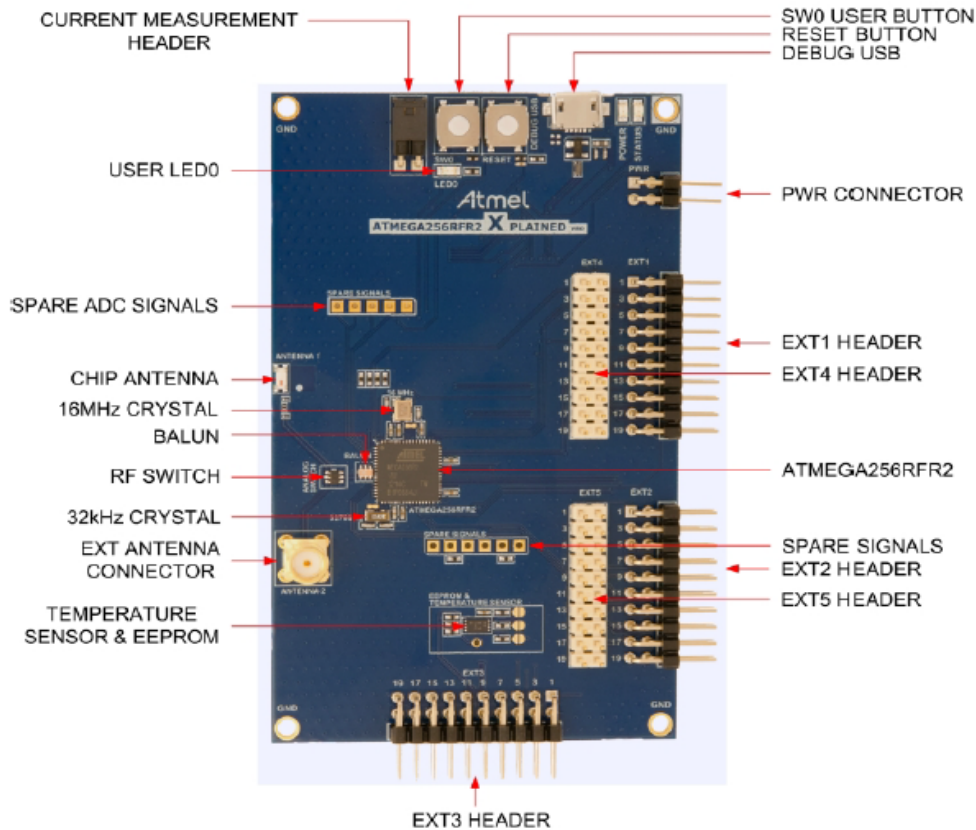


Figure 3.2.1: AccuTag Display Unit, ATmega256RFR2 Xplained pro board[4]

The next component is the E-ink display extension kit (EPD), which is manufactured by the Pervasive Display Company. These two components are integrated through the extension connectors that are available on both the boards.

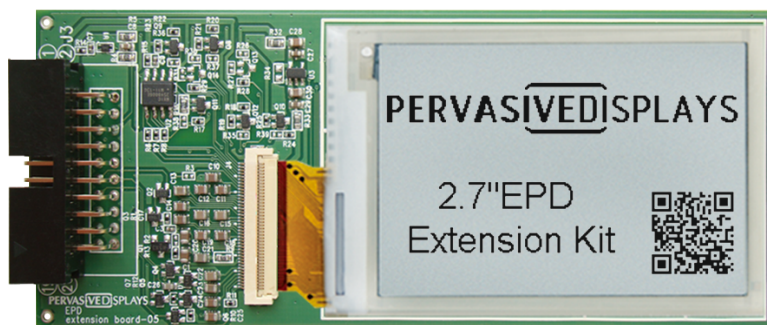


Figure 3.2.2: AccuTag Display Unit, E-Paper Display (EPD)[5]

The following figure illustrates the hardware block diagram of the EPD board. The display will be controlled via the SPI connection, by the ATmega256RFR2 chip. The ATmega Xplained Pro board has 5 available extension headers for connecting additional components depending on the application demand. For the AccuTag display prototype we will be using extension header 1, shown

is figure 3.2.4. The two 20-pin extension connectors available on both boards and their respective connections are also tabulated in table 3.2.1.

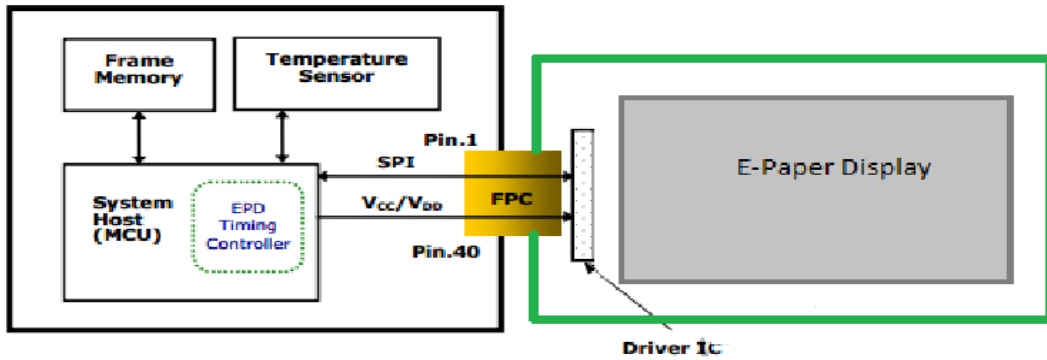


Figure 3.2.3: AccuTag Display Unit, E-Paper Display (EPD) block diagram[5]

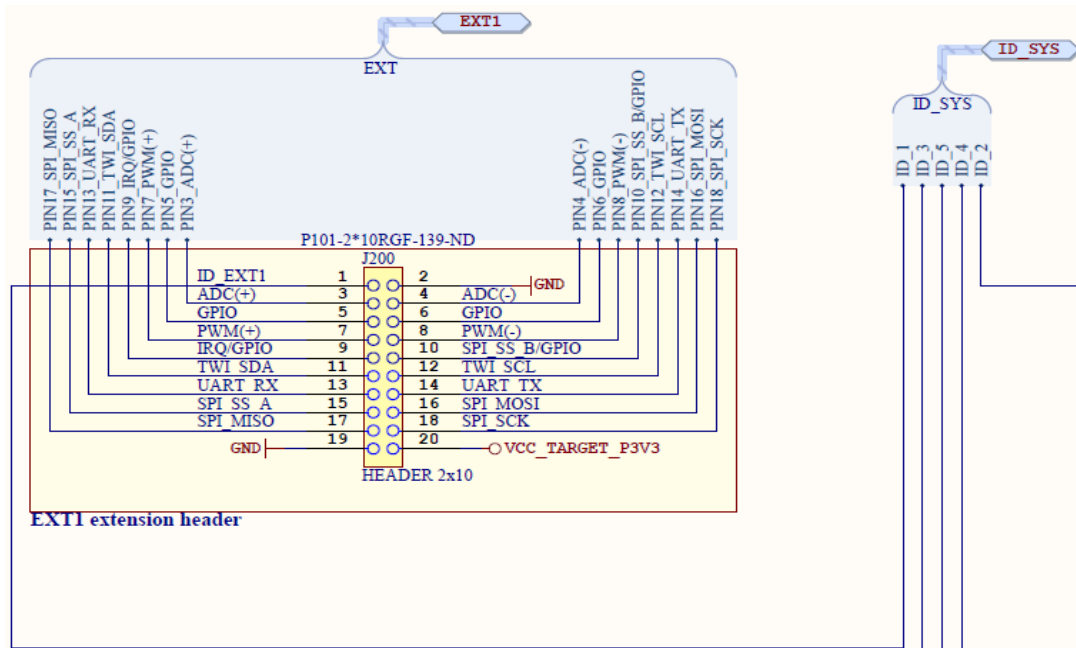


Figure 3.2.4: AccuTagXplained Pro extension header[6]

EPD kid Extension pin #	X-Plained Pro EXT1 pin #	ATmega256RFR2pin	Function
Not connected	# 1_SYS ID		
Not connected	# 2_GND		GND
# 6	# 3_ADC0	# 63_PFO	Temperature
# 13	# 4_ADC1	# 64_PFO	Boarder control
# 12	# 5_GPIO	# 48_PE2	Discharge
# 10	# 6_GPIO	# 49_PE3	Reset
# 9	# 7_OC1A	# 41_PB5	PWM (+)
# 11	# 8_OC1B	# 42_PB6	Panel on_PWM (-)
# 8	# 9_GPIO/INT5/IRQ	# 51_PE5	Interrupt request line (busy)
# 18	# 10_SPI chip select B	# 30_PD5	Flash chip select
Not connected	# 11_TWI SDA	# 26_PD1	Data line for two wire interface
Not connected	# 12_TWI SCL	# 25_PD0	Clock line for two wire interface
# 3	# 13_UART0 RXD	# 47_PE0	Receive line for UART
# 4	# 14_UART0 TXD	# 46_PE1	Transmitter line for UART
# 19	# 15_SPI chip select A	# 14_PG0	EPD chip select
# 15	# 16_SPI MOSI	# 38_PB2	SPI MOSI
# 14	# 17_SPI MISO	# 39_PB3	SPI MISO
# 7	# 18_SPI SCK	# 37_PB1	SPI clock
# 20	# 19_GND		GND
# 1	# 20_VCC		VCC

Table 3.2.1: Extension connector pin layout table[5]

The top-level block diagram of the AccuTag Display unit is shown in the appendix (figure A2). Furthermore it is followed by detailed schematic of the ATmega256RFR2 Xplained pro board, antenna and the dual input power supply used in making the AccuTag Display unit.

3.3 AccuTag Display PCB layout considerations

Please note following specifications will be taken into account when making the final product and upon manufacturing.

3.3.1 Mechanical Specification

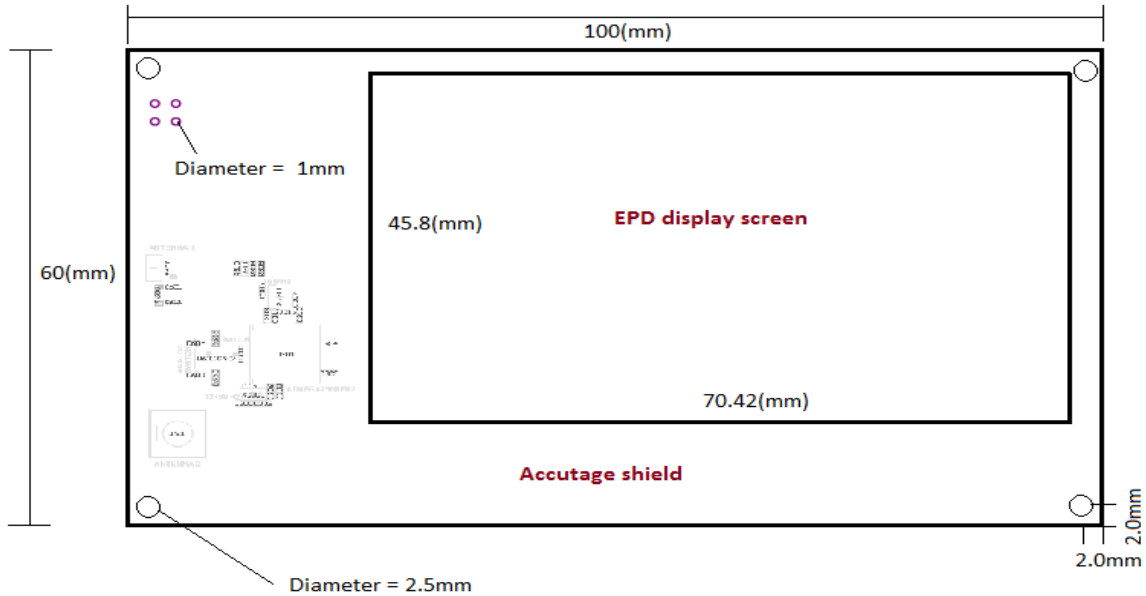
As per mechanical specifications, we made certain choices for the EPD hardware design based on resource availability and operational constraints. The EPD display tag is to be permanently affixed to the PCB board containing the Atmel microcontroller ATMEGA256RFR2 and other electronic components.

Table 3.1.1 below details the dimensions of our Accutag display shield.

Board Dimensions	Value	Uncertainty (%)	Unit
Horizontal	100	1	mm
Vertical	60	1	mm
Thickness	1.2	10	mm

Table 3.3.1: Accutag Display dimension parameters

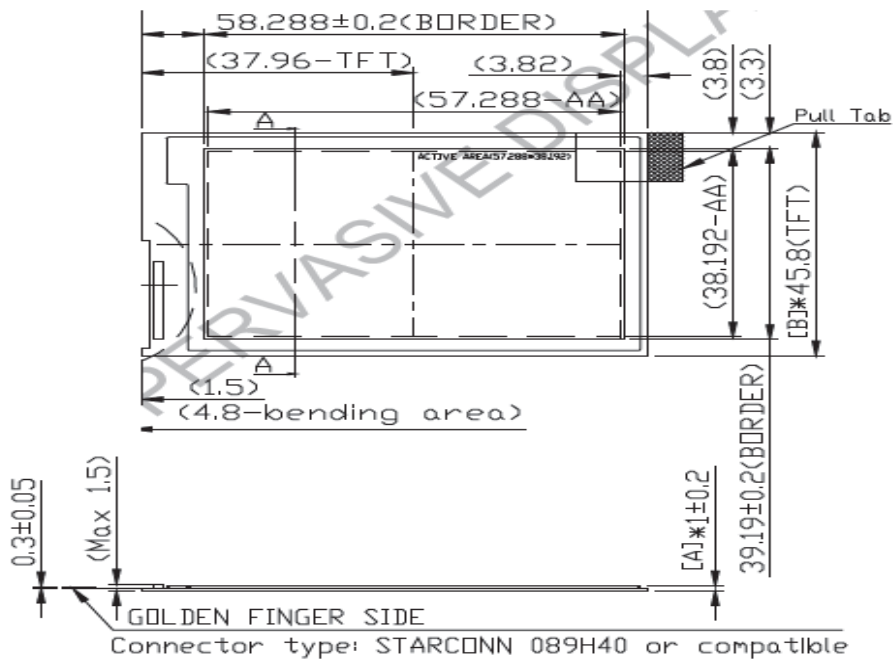
Figure 3.3.1 below illustrates the mechanical dimensions of our final product.



Figure

3.3.1: AccuTag Display top-view drawing

For further details, provided below is the EPD display drawing provided by Pervasive Displays Inc. The display is to be epoxy bonded to the PCB board to avoid falling off the screen. Our design is based on running tracers of width 0.5 mm (51 thou) between the display and the shield instead of the FPC connector.



General tolerance: $\pm 0.3\text{mm}$

Figure 3.3.2: EPD screen dimension details provided by Pervasive Displays Inc[7]

Some miscellaneous mechanical specifications are listed below in table 3.3.2.

Board Spec	Value/Type	Uncertainty (%)	Unit
Weight	30	15	g
Surface Area	322.4	5	mm
Layers	1	-	-
EPD Trace pitch	51	5	thou
IC leg spacing	0.02	1	thou
Grid through hole size	100	1	thou
Grid tracking	50	1	thou
Mount Hole sizing	2.0 (diameter)	1	mm
Standard Via sizing	0.5 (standard)	1	mm
Pad sizing	1.8 times hole diameter	1	-
Core type	FR4	-	-

Table 3.3.2: Mechanical requirements

3.3.2 General Specification

Wiring Layout

For our PCB trace wiring, we will be avoiding the daisy chain wiring scheme. This is a precautionary measure aimed to avoid the power shutdown and circuit breakdown due to the failure in one or more electronic circuit components. The preferred wiring scheme for our board is the parallel wiring scheme as shown in the figure below.

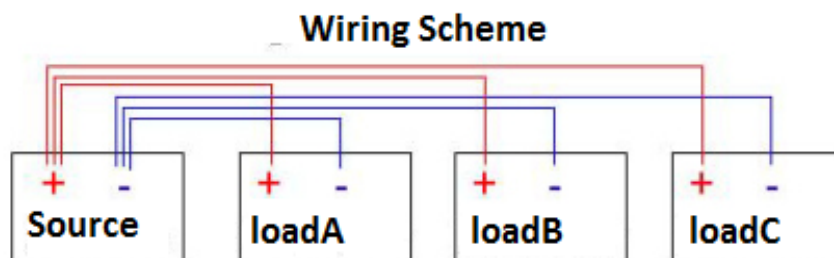


Figure 3.3.3: Wiring layout schematic for component placement

Placement of components Digital vs. Analog power Grids

Digital and Analog circuits have to be powered via separate power pins. Decoupling capacitors are to be placed close to the power connector pins in order to isolate the digital and analog supplies. The bypass capacitor will help the AC ripple caused by DC sources to dampen by tracing it to the ground. Bypass capacitors of range 0.01 to 0.1 μF are typical values to be used between VDD sources and ground pins [9].

Thermal Expansion

The high soldering temperatures can become a concern when dealing with PCB design. The amount of expansion of the material can be measured in terms of a value known as coefficient of thermal expansion (CTE). Copper used in the PCB design has a typical CTE value of 18 (measured in ppm (parts per million) per °C) at 20°C. The adhesive material such as Epoxy resin has a CTE of 55 which means we are likely to encounter soldering mismatch problems. To cope with this problem, we use woven glass cloth which provides strength and flexibility to epoxy resin in XY axis, bringing down its CTE to the same level as that of copper. We still make sure the Z axis CTE is not a problem, considering the height of various components on the PCB [8].

Hole Sizing Standards

For our PCB design, we follow “Rule of 10” according to which small vias will have the hole size of 10 mils with additional 10 for pad rings. Also, traces should be laid approximately 10 mils further from the edge of the board. For larger vias (40-60 mil) diameter, the holes should also have pad rings of 1.8 times the diameter for clearance beyond the component sizing. [8]

Environmental Specification

The temperature range for the AccuTag shield storage is -40°C to 125°C. The operational ambient temperature is 0°C to 50°C.

Electrical Specification

Below in the table are the electrical ratings for the operation of the AccuTag system’s transmitter and price-tag unit.

Parameter	Symbol	Value			Unit
		Min	Typ	Max	
Digital Power	V_{DD}	1.8		3.6	V
Analog Power	V_{CC}	1.8		3.6	V
Power Output	P_{out}		3.5		dBm
Receive Current Consumption	P_r		12.5		mA
Transmit Current Consumption	P_t		14.5		mA

Table 3.3.3: System Electrical Requirements

4 Software Design

There will be software running on all components of the AccuTag system. Software to transmit control signals from the host computer to the transmitter, software to broadcast RF signals from the transmitter, software to program the e-ink display etc. As such it is important to clearly specify the design for each piece so that integrating them together can happen efficiently, seamlessly and without any issues. The following section first starts by providing a high level view to show how the components of the AccuTag system are connected by software and what each one's role is. Then further detail about the design of each individual component will be given until software design for the entire system is given and no ambiguity remains.

4.1 High Level Design

The following is a high level figure of all the software components of the AccuTag system and brief description of what each one is responsible for.

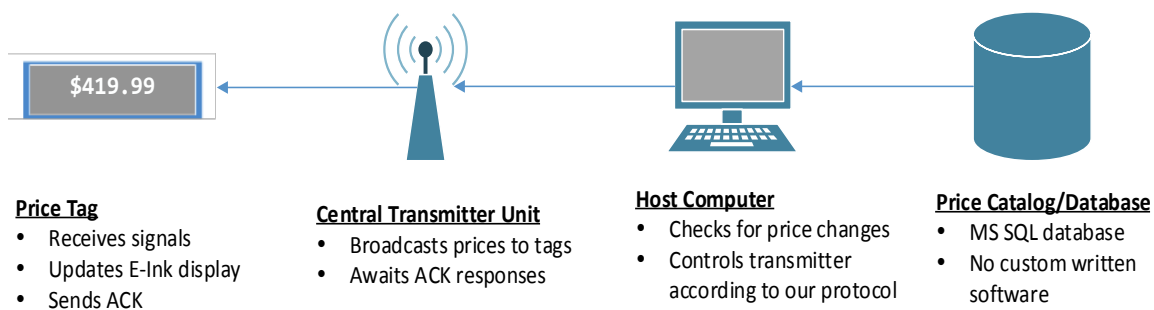


Figure 4.1.1: High Level View of the AccuTag system

Price Catalog/Database - the software here is not written by the Precision Wireless team. It is simply a Microsoft SQL server that's been setup with the proper table schema and is accessible by the host computer.

Host Computer - this component's role is to check for updated prices in the database then use some APIs to relay this information to the transmitter. This and the transmitter communicate according to a simple protocol discussed later.

Central Transmitter Unit - the program on this device waits for new messages from the host computer then broadcasts a new price update accordingly.

Price Tag - these units await new packets from the transmitter. If the packet is meant for them they parse the packet and use it to update the e-ink display.

4.1.1 Database

The database is a catalogue on a Microsoft SQL Server Database. This catalogue will have a single table to store the price of each item in. Refer to the regular table below for details on what is contained in the database table.

Column Name	Data Type	Example Value
tag_id	int	25
item_name	nvarchar(255)	"Welch's Gummies"
item_price	money	1.79
item_info	ntext	"2 For 1 Sale"

Table 4.1.1: Price catalogue's table schema

Note that ntext and nvarchar are used (as opposed to text and varchar) so that Unicode characters may be used to display messages in other languages.

4.1.2 Host Computer

The host computer's software will be the intermediary between the transmitter and the database. Our aim is to allow the end users of the AccuTag system to write programs for their own stores to suit their needs. So rather than providing a single program to all our customers, we instead opted to write a set of basic APIs and a small program that will allow for barebones operation of the system.

We decided to use C# to write the software on the host computer because the .NET framework allows us to rapidly develop our programs with an extensive library full of numerous different classes and because the same framework provides libraries for communicating easily with SQL Server databases. Unfortunately, as a result, this means our software will only be available for the Windows platform. Porting our software to *nix platforms will not be challenging and this is planned for future versions. The only challenging part is finding drivers that interface between *nix operating systems and Microsoft's SQL servers. FreeTDS[ref] is a good candidate for such a driver.

4.1.3 Protocol

Communication between the host computer and the transmitter is done via USB through a serial COM connection. The protocol we came up with is very basic and uses fixed packet sizes; speed will not be significantly effected at all. Each packet sent by the host software has the following structure.

Name	Size (Bytes)	Offset	Description	Example
Packet Type	1	0	Indicates whether this signal is to update a price or retrieve a price.	“U” – update “R” – retrieve
Tag ID	2	1	Identifier of the price tag.	0x0019
Item Name	16	3	Name of the product.	“Welch’s Gummies”
Price Digits	5	19	Single byte digits of item price. Decimal point is placed before last 2 digits.	0x0000010709 (1.79)
Item Info	32	24	Misc. information related to the item.	“2 For 1 Sale”

Table 4.1.2: Host to transmitter message format

Note: each field can be left as all 0s to indicate that it should not be changed or if the fields aren't necessary (e.g. in case of a retrieve operation). The price tags are programmed to ignore these kinds of fields when displaying new information on the e-ink displays.

The total size of each packet is fixed at 56 bytes.

4.1.4 API Set

The API are designed to be as basic as possible so that the end user can use them as they see fit. Our aim wasn't necessarily to make it easier for them to write their software but rather to not restrict them in the ways they can write it.

These APIs communicate with the transmitter via a serial COM connection. Hence why each one takes a SerialPort object as an argument. Each API has been written with our protocol in mind. This means that each call to the API will always send 56 bytes to the transmitter. Each one returns an 8-bit integer indicating any errors that may have occurred or a 0 to indicate success. An exhaustive list of error types has not been determined as development on this portion of the system has not begun yet.

The table below describes each of the three APIs.

API Name	Arguments	Description	Return Value
UpdatePriceByID	com: SerialPort object ID: 16-bit integer Price: 5 byte array	Update price by tag ID	8-bit integer indicating any errors
UpdateNameByID	com: SerialPort object ID: 16-bit integer Name: 16 byte array	Update item name by tag ID	8-bit integer indicating any errors
UpdateInfoByID	com: SerialPort object ID: 16-bit integer Info: 32 byte array	Update item information by tag ID	8-bit integer indicating any errors

Table 4.1.3: List of basic APIs

4.1.5 Central Transmitter Unit

The software running on the transmitter is built on top of the framework provided by Atmel. This framework came with an example project called a 'Performance Analyzer'. We stripped away the unnecessary parts and used the underlying framework to build our program.

The main routine of the program sits inside a while loop and waits for new data in the COM port. The logic for this routine is illustrated in the following flowchart.

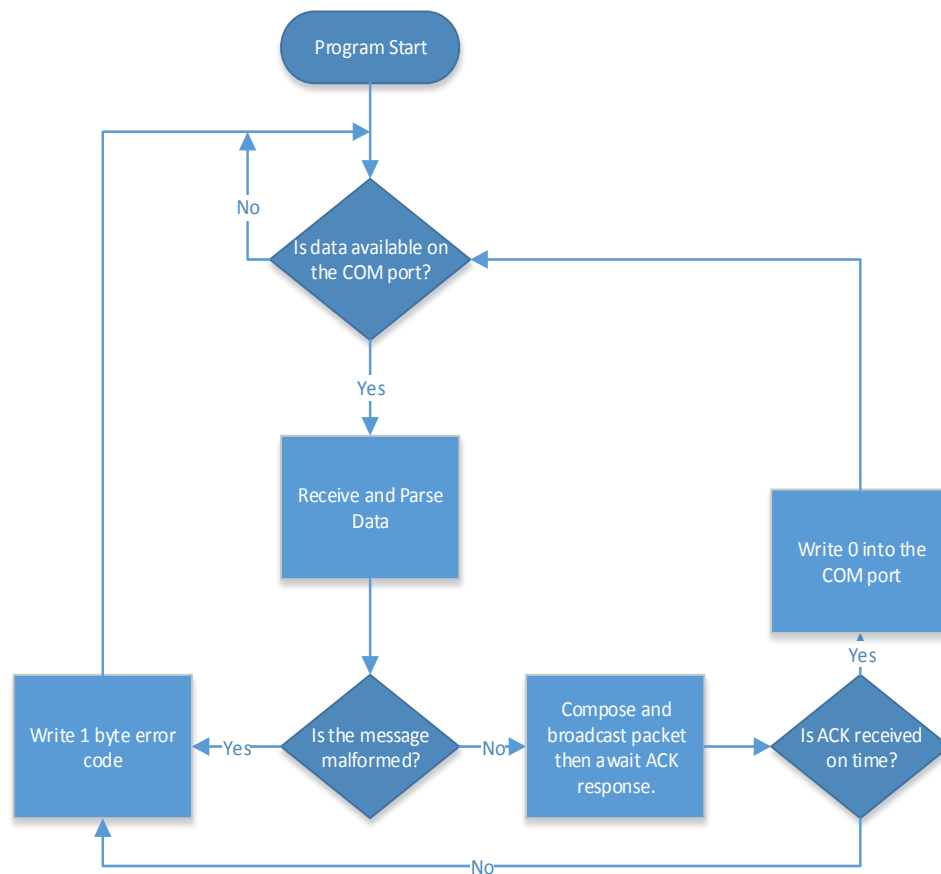


Figure 4.1.2: Flowchart for the transmitter unit's program logic

The 5 second wait time for an acknowledgement will suffice for our rat's-net prototype but is not ideal for a production system since it could take a very long time to update many prices. Production versions will need to rely on a multithreaded design so that the acknowledgements can simply be received and placed in a queue (by another thread) then written into the COM port.

4.1.6 Price Tag Units

The program running on the price tags has the most simply logic. It basically waits for packets, parses the packets then displays information on its display. Nevertheless, the flowchart below provides more detail in on this component's logic

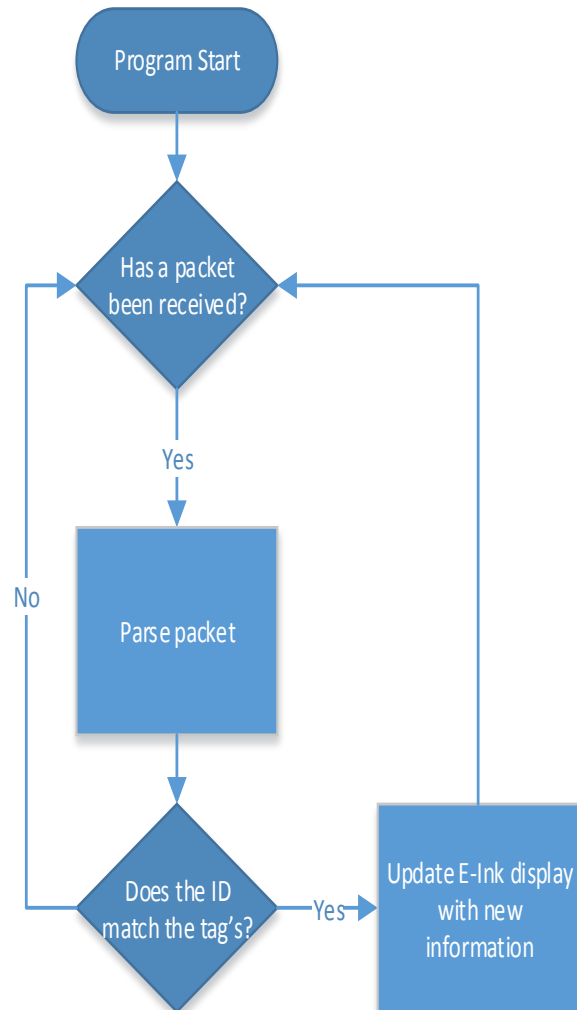


Figure 4.1.3: Flowchart for the price-tag unit's program logic

4.2 E-Ink Display

This section contains detailed information in the design of the timing controller for the 2.7" Pervasive Displays E-Ink screen. The E-Ink display is operated by a COG (chip on glass) driver that is interfaced based on a microcontroller program in order to display images using two pages of memory buffer.

In general, new and previous display images are stored in the memory buffer until the COG driver is powered on and initialized. The E-Ink screen then updates in a sequence of stages until the new image is completed, at which point the COG driver will then power off.

The flowchart below provides an overview of the update cycle necessary for the E-Ink display (also known as EPD by Pervasive Displays) to display a new image. The SPI module is explained in section 1 which is required for all other sections. Each component of the block diagram will be further explained in their respective sections. All of the following flow charts were implemented through the use of programming in high level C coding using AVR Studio 6.1.

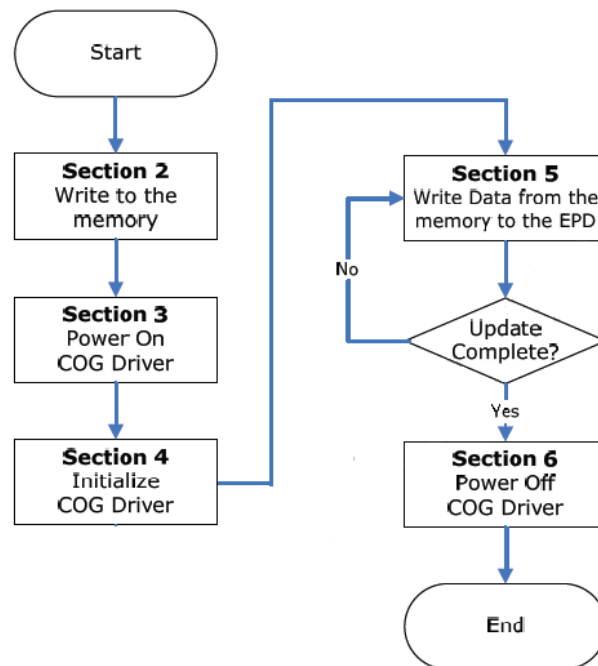


Figure 4.2.1: Overview of Update Cycle for E-Ink Display

4.2.1 SPI Timing Control

In order to have the E-Ink display properly update, there are a list of control signals which must be activated at specific times. The diagram shown below depicts the signal control overview during the whole update cycle. As correlated with the flow chart above, different sections require different signal configurations.

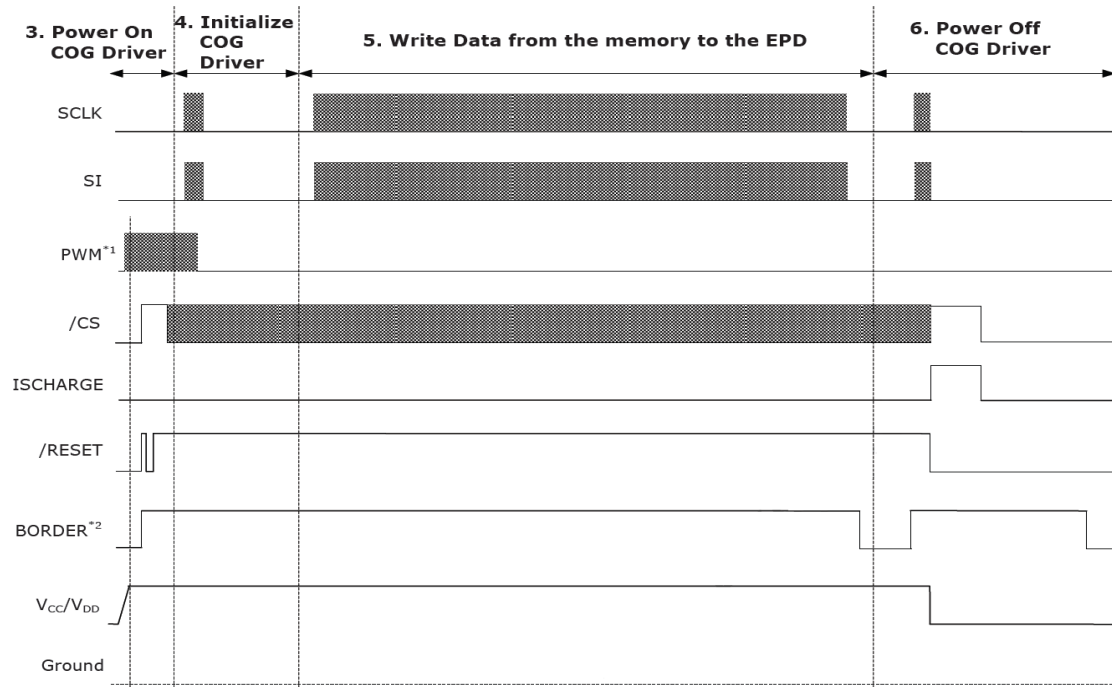


Figure 4.2.2: Overview of Control Signals during Update Cycle of E-Ink Display

SPI commands are used to communicate between the Atmega256RFR2 Xplained Pro microcontroller and the COG driver. In terms of the standard method of SPI format, this E-Ink display uses a one way communication and follows the above waveform in terms of the control signal (CS) in order to manage the update cycle. The maximum clock that the display can accept is 12MHz. Since the only clocks present on our MCU (ATmega 256RFR2 Xplained Pro) are 16MHz and 32MHz, we will use clock division to attain the 12MHz clock (ie. multiplying the 16MHz clock by 3/4).

Based on research, the SPI format for the Pervasive Displays E-Ink screen is predetermined in the following way:

SPI(0xI₁I₂ , 0xD₁D₂D₃D₄,D₅D₆D₇D₈...)

Where

- I_mI_n is the Register Index and the length is 1 byte.
- D_{m~n} is the Register Data. The Register Data length varies from 1, 2, to 8 bytes depending on which Register Index is selected.

In order to send SPI commands to the E-Ink screen, the microcontroller interface must first program the SPI (shown as SI in the figures) to send a 0x70 header command before sending the register index. There is a similar procedure in which SPI (SI) must send a 0x72 header command prior to sending the register data. The control signal (CS) must also be triggered to set for High then Low between the register index and register data. The follow waveform and flowchart shows the full procedure for sending SPI commands.

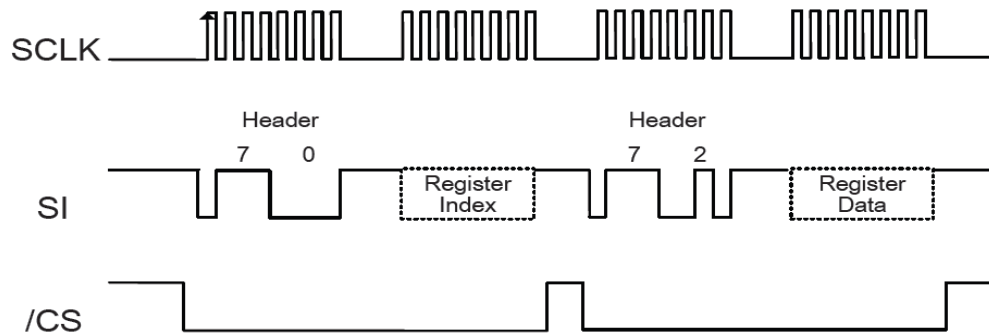


Figure 4.2.3: Timing Waveform for sending SPI commands using SCLK, SI, and CS

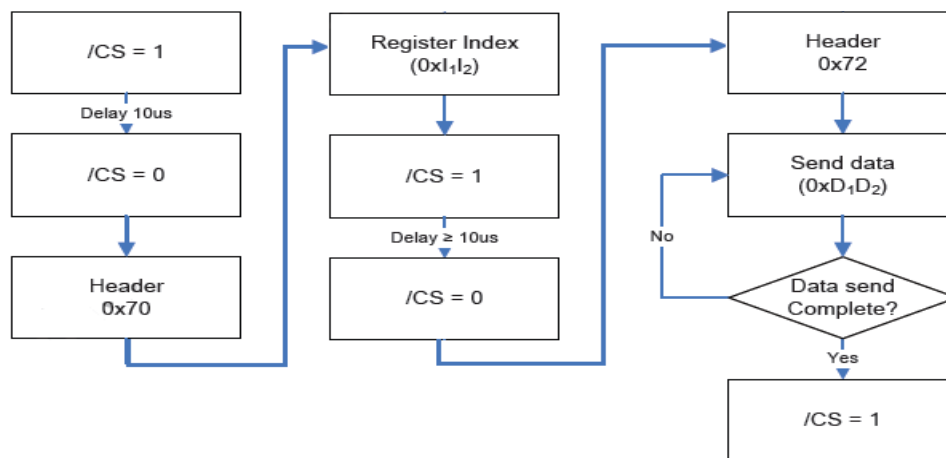


Figure 4.2.4: Procedure for sending a generic SPI command (ie. SPI (0x11I2, 0xD1D2))

4.2.2 Writing to Memory

Data must be stored in memory in the form of a single-bit bitmap format (black/white) before powering on the COG driver and writing to the screen. This new image is stored in an image buffer along with the previous image in another. Once the COG driver is powered on, there will be a comparison between the two images before an update to the E-Ink screen is made. The available buffer space for both the previous and new image buffer as determined by Pervasive Displays for the 2.7" E-Ink screen is approximately 11,616 bytes. The amount of space is much greater than required for displaying product price and information as only a couple of bytes will be required.

4.2.3 Powering on the COG Driver

The following block diagram depicts the full procedure for powering on the COG Driver.

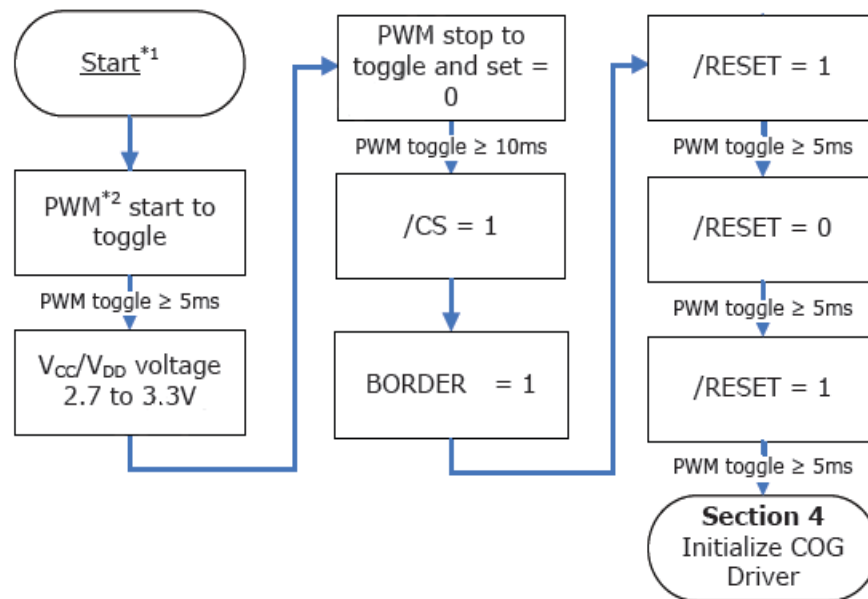


Figure 4.2.5: Flow Chart for procedure on powering on the COG Driver

In the initial Start phase we set the supply voltages (V_{CC}/V_{DD}), for the analog and digital part of the source driver respectively, along with control signals RESET, CS, BORDER, SI and SCLK all to zero. At this stage we have already programmed the SPI module for sending commands along with an image of the new and previous pattern in the SRAM. We begin by toggling the PWM using a 100-300KHz, 50% duty cycle, square wave and continuing the toggling process throughout the whole power on procedure. The process is fully implemented through the use of programming in order to manipulate each control signal along with the PWM toggling at exact times in order to reduce delays.

4.2.4 Initializing the COG Driver

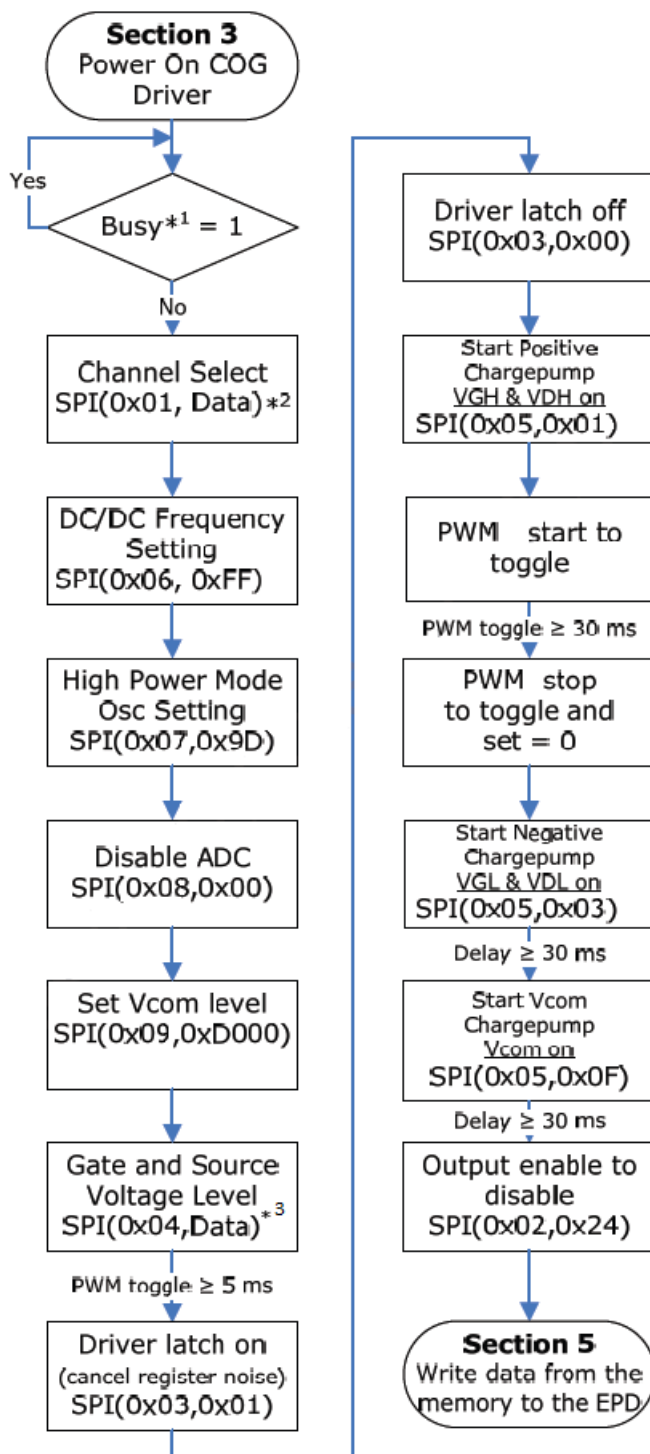


Figure 4.2.6: Flow Chart for procedure on initializing the COG Driver

After the COG driver is fully powered and operational, initialization steps as required based on the documentation given for the E-Ink display.

Key notes regarding the block diagram on the left are as follows.

[1] A busy signal was not implemented as the AccuTag system is designed to be automatic without any feedback analysis. This helps in efficiency and convenience even in the design stages of development. Instead, a delay signal of approximately 1usec was used in order to ensure that the COG driver will have sufficient time to process the previous input data before receiving new data from the SPI.

[2] The SPI(0x01, Data) command used for our 2.7" screen is:

SPI(0x01, (0x0000,007F,FFFE,0000))

[3] The SPI(0x04, Data) command used for Gate and Source Voltage Level for our 2.7" screen is:

SPI(0x04, 0x00)

4.2.5 Writing Data from the Memory to the E-Ink (EPD) Screen

This section describes how data should be sent to the COG Driver which will update the display. The COG Driver uses a buffer to store a line of data and then writes to the display. The 2.7" E-Ink Display has a resolution, in terms of pixels, of 264 (width) by 176 (height). The data structure of the pixels are further organized into data components as follows:

- One Bit - Can be either White (W), Black (B) or Nothing (N). The use the nothing bit is for mitigating ghosting when refreshing and updating the E-Ink display
- One Dot/Pixel - Comprised of 2 bits. These two bits must be of the same type.
- One Line - The number of dots or pixels in one full horizontal line on the E-Ink display. For the 2.7" E-Ink Display, this equates to 264 dots/pixels.
- One Frame - The combination of lines (horizontal) and rows (vertical). For the 2.7" screen, one frame of data equates to 176 lines * 264 dots. The MCU Frame Time which is used is approximately 80ms.
- One Stage - The number of frames used to write an intermediate pattern or image. The 2.7" display requires 3 frames of data per stage with a stage time of about 630ms.

These terms will be important in understanding the flowchart below which will describe the full process required for sending data to the COG Driver along with writing the data to the display.

The process of writing to the display is done in four stages. A standard example of updating a previous image to a new image is shown with their respective four stages below.

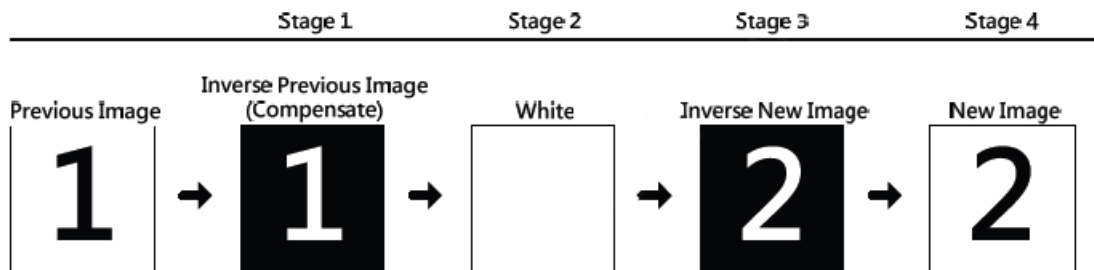


Figure 4.2.7: Example for writing to the E-Ink display in four stages

Once implementation of the programming at this lower level is complete, there requires a predefined library of images for letters and numbers which will be used to update the E-Ink display. This allows the freedom of specifying sizes specific to the needs of our design specifications. Furthermore, defined functions in terms of high level coding regarding price change, product name change, product information change and barcode change will also be implemented so that it will allow a easy learning curve for market users.

The following flow chart describes the details of how to send data to the COG Driver. The COG Driver then uses a buffer to update the display line by line. This was taken from the documentation for the 2.7" Pervasive Display.

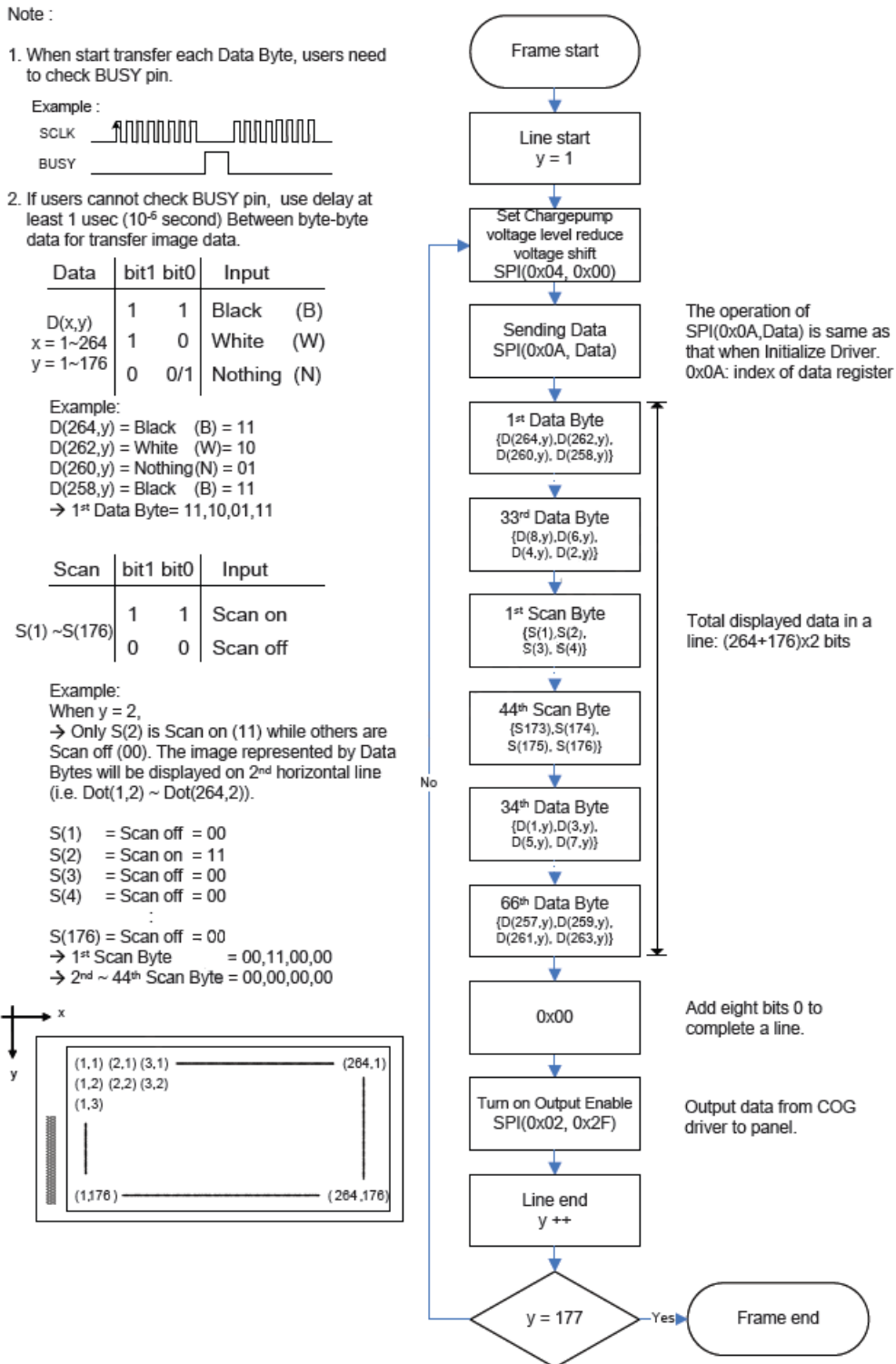


Figure 4.2.8: Flow Chart outlining full procedure on how to send data to the COG Driver

The following flow chart contains the method for displaying the image in stages as mentioned in the figure above. Each of the four stages should contain the same number of frames which are rewritten during each stage. This flow chart indicates the procedure for updating an image from the previous displayed image stored in the memory buffer to the new image.

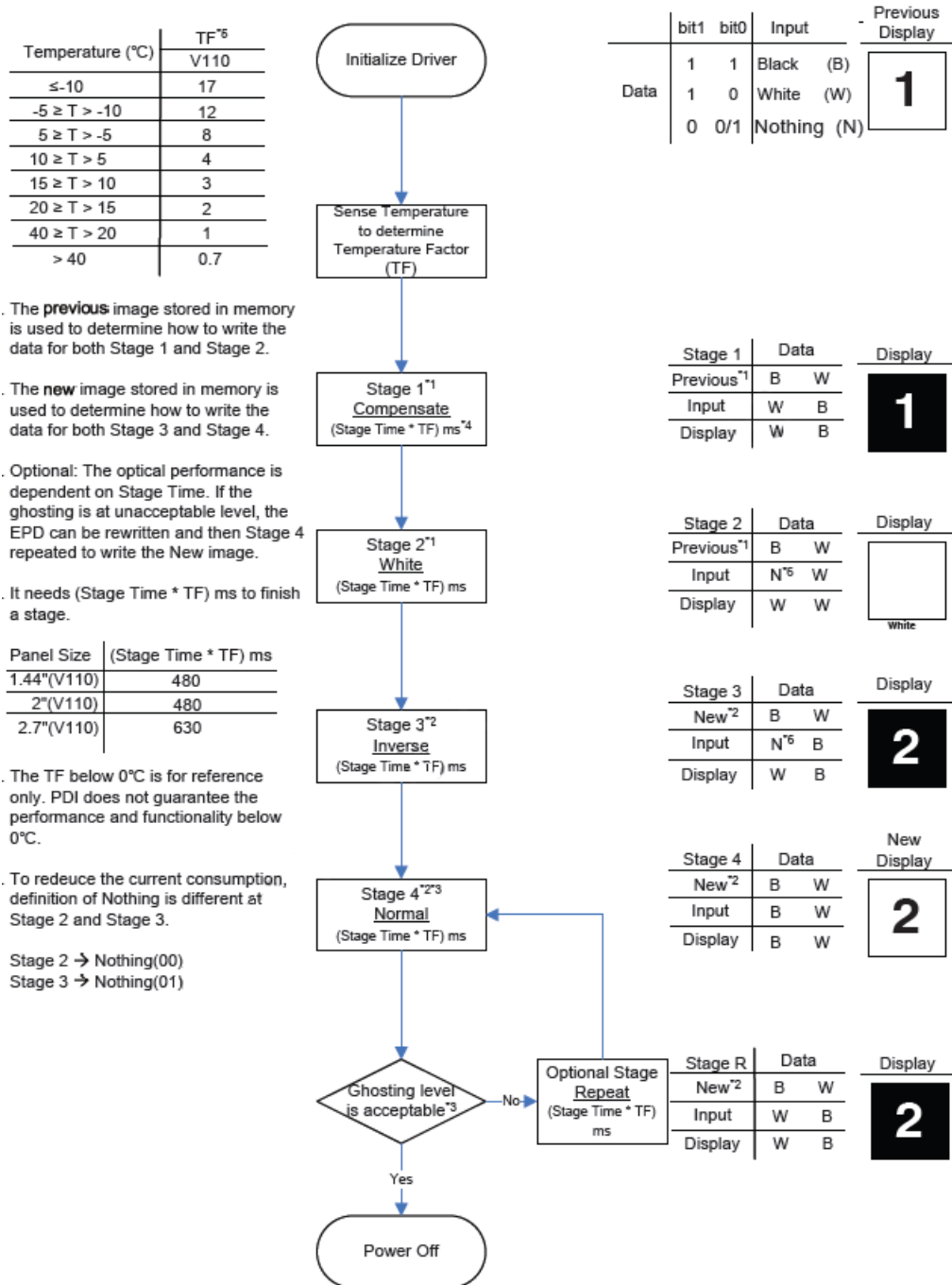


Figure 4.2.9: Flow Chart outlining full procedure on how to display data to the E-Ink Display using four stages

4.2.6 Powering off the COG Driver

After the E-Ink screen has been successfully updated. The COG Driver must be powered off in order to reduce power consumption. As this was one of the key components of the AccuTag's initial design specifications, this step is crucial in ensuring that our product fits the criteria on which it was first created. The following flow chart outlines the SPI commands and control signal commands which must be programmed in order for the COG Driver to go back to its low power state and await the next E-Ink display update.

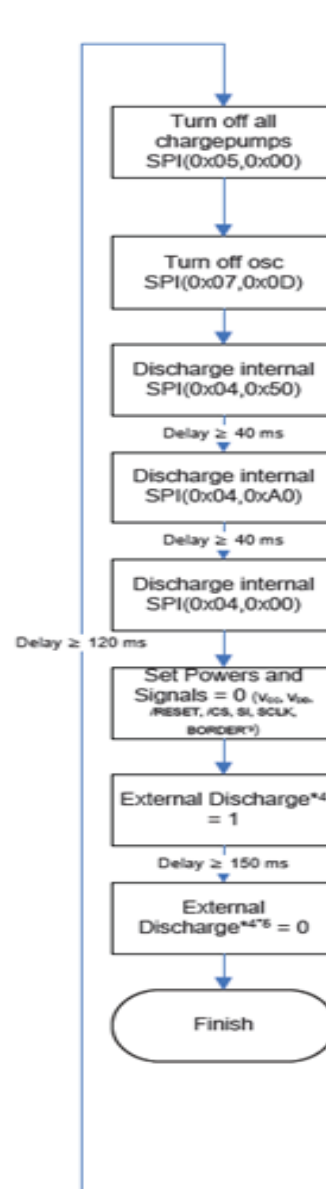


Figure 4.2.10: Flow Chart outlining full procedure for powering off the COG Driver

5 System Integration / Test Plan

5.1 Test Plan

Precision Wireless has developed a set of test procedures for testing device functionality as well as the reliability and usability of the AccuTag system. Since, Precision Wireless is in its early stages of device development, the test procedures prescribed are not final and are subject to technical review given the expected change or adaption to different integration deadlines and requirements. The test plan for the various components of the AccuTag System is discussed in the following sections.

5.1.1 Database Test Plan

The database access test procedure will primarily focus on two aspects, namely the access speed and reliability. A test script will be written to conduct queries on a set of data types (refer to table 4.1.1) and then ensure the time it takes the query to be answered. An average of the access time for each data type is performed. The average access time should not exceed 3 seconds which is the maximum accepted latency of our database system and no data corruption is acceptable. Note, however, that this time is subject to some leniency if the database is accessed over the internet, in which case it is up to database admins to keep it at a manageable level that they see fit.

The reliability of the system will be measured through endurance testing which would involve writing a script to periodically query information from the database for a 24 hour period and the endurance capability of the database system is measured based on average access speed during this time period. For ensuring that the database is highly reliable no outages are acceptable.

5.1.2 Transceiver Test Plan

The transceiver combination, i.e. the process of communication between the transceiver in the price tag and the transceiver at the central transmitter unit, will be tested for functionality and reliability. The functionality of the transceiver combination will be tested for characteristics such as range, Packet Error Rate, Sensitivity and Output Power. All the tests for the transceiver will be performed with a Performance Analyzer Application provided by Atmel. The Performance Analyzer Application is an application module built on top of the TAL (Transceiver Abstraction Layer) of Atmega256RFR2. The state machine for the Performance Analyzer Application is given below.

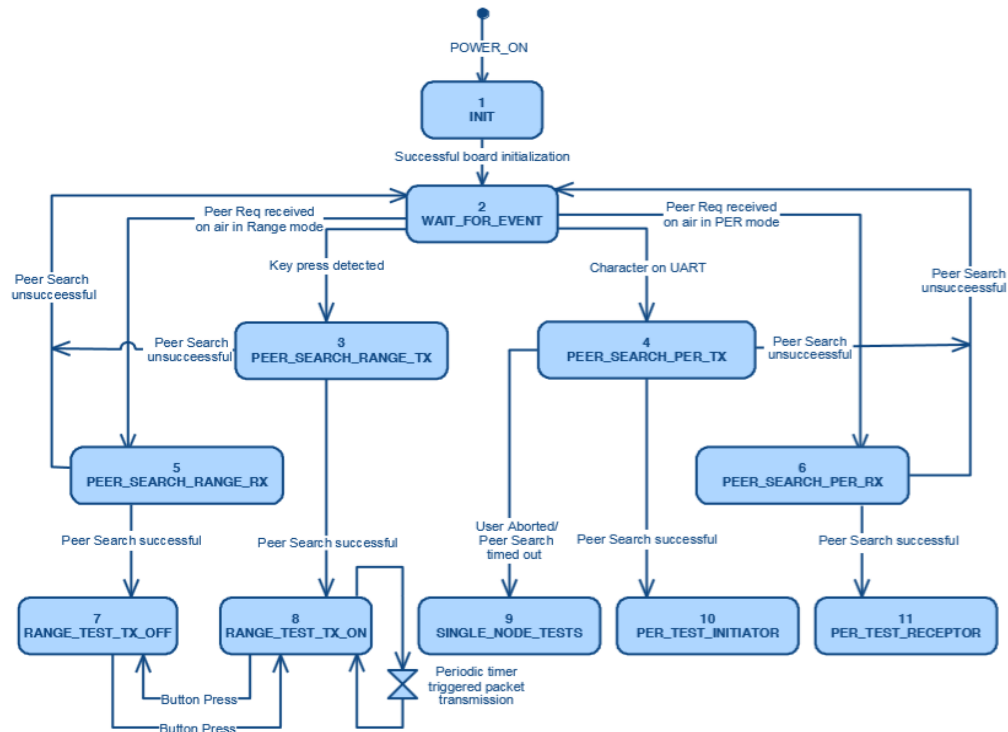


Figure 5.1.1: State machine of the performance analyzers

5.1.3 Range Testing

For testing the range of the transceiver combination, we shall use the transmitter and the receiver boards programming with the Performance Analyzer. Since, the receivers can confirm reception of packets with an LED (Rx LED) and the transmitter can confirm packet transfer through the use of its own LED (Tx LED), the tester for appropriate testing of range shall perform the specific tasks as listed below.

1. Power on the two boards i.e. the transmitter board and the receiver board.
2. Program the boards with the Performance Analyzer application code provided by Atmel.
3. The tester then shall press the push button on the transmitter node to enable range testing and should observe Tx LEDs flashing on the transmitter and the Rx LEDs flashing on the receiver indicating successful transmission and reception.
4. First, the tester shall confirm that the LEDs are flashing at a distance of less than 1m away before moving on the next step.
5. This step involves two testers. The two testers should hold the two wireless nodes and shall incrementally increase the distance between them. Furthermore, the testers also need to confirm no presence of any obstructions between them.
6. The testers will now start to incrementally move away from each other with the boards in hand and will confirm at each new distance of separation that the LEDs are flashing. The test will continue until the LEDs stop flashing and the distance between them is recorded.

User Input - Pressing of the Tx push button to enable range testing mode.

Failure Conditions -The range test shall fail if the LEDs stop flashing after pressing the Txpush button at a distance of separation less than 100 meters between the testers.

Success Condition -The Range test shall fail if the LEDs keep on flashing after pressing the Tx push button at a distance of separation of 100 meters or more between the testers.

In such manner, the test determines the maximum range the transceiver combination can communicate across. A range of 100 meters is considered the minimum threshold by Precision Wireless for successful test completion of the AccuTag system.

5.1.4 Packet Error Rate(PER) Testing

The Packet Error Rate of the transceiver combination is determined with the help of the Performance Analyzer Application according to the IEEE 802.15.4 standard. The test requires the completion of the following steps:

1. Power on the two boards (transmitter unit and price-tag/receiver unit) and program them with the Performance Analyzer Application.
2. The transmitter board should be connected to a machine with a terminal window so that the tester and see use standard IO (e.g. seeing output and entering input through the keyboard). Atmel's AVR Studio IDE comes with a Terminal Window plugin that is perfectly suited for this.
3. Then any character can be typed in the terminal window which triggers the Tx into the PER testing mode. In this mode, the terminal window will continuously update information on the five parameters: number of packets transmitted, number of packets successfully received, Output power, Sensitivity and the PER. The tester shall record the PER when 2000 packets have been transmitted.

User Input - Typing a character in the terminal window to trigger PER testing mode.

Failure Conditions -The PER test shall fail if the PER is more than 0.016 a rate determined for IEEE 802.15.4 standards.

Success Condition -The PER test shall succeed if the PER is less than 0.016 a rate determined for IEEE 802.15.4 standards.

5.1.5 Sensitivity and Output Power Testing

The Sensitivity and Output Power of the transceiver combination is also determined with the help of the Performance Analyzer Application according to the IEEE 802.15.4 standard. The test procedure is identical to that of PER testing described above, except that at the end of the test the Average Output Power and Average Sensitivity are recorded.

User Input - Typing a character in the terminal window to trigger PER testing mode.

Failure Conditions - The test shall fail if either the output power is less than 3.5 dBm or if the sensitivity is less than -100 dBm.

Success Conditions - The test shall succeed if the output power is more than 3.5 dBm and the sensitivity is more than -100 dBm.

5.1.6 Reliability Testing

The reliability testing for the transceiver combination would involve endurance testing. Endurance testing follows the same basic procedure prescribed by PER and Power Output testing except it will be run for a continuous 12 hour period and at the end the following five parameters will be recorded: number of packets transmitted, number of packets successfully received, PER, average output power, and average sensitivity.

User Input - Programming the Tx and the Rx boards with the code provided and recording the five parameters as specified above.

Failure Conditions - The test shall fail if either the output power is less than 3.5 dBm or if the sensitivity is less than -100 dBm or if the PER is more than 0.016.

Success Conditions - The test shall succeed if both the output power is more than 3.5 dBm and if the sensitivity is more than -100 dBm and if the PER is less than 0.016.

5.1.7 Microcontroller Testing

Microcontroller testing of the AccuTag system will involve testing the firmware. The microcontroller on the transmitter unit and the price-tag unit differ in the programs they execute so they will be test separately.

The main purpose on the microcontroller on the transmitter unit (i.e. the program described in figure 4.1.2) is to receive messages from the host computer then use it compose packets for broadcasting. Testing this program's functionality is best done by sending bad messages and confirming their rejection then sending proper messages and confirming that they are broadcast correctly.

The following is a more detailed description of this test but note that since development has not begun on this portion of the software the test procedure is not as detailed. But a clear picture of the desired end result is given.

1. Program the transmitter node with the transmitter program written by Precision Wireless.
2. Use the host computer software to send a series of bad messages to the transmitter (i.e. non alpha-numeric characters, packet is too small or too big, first byte isn't 'U' or 'R' etc.)
3. Confirm by log output that each of these messages has been rejected.

User Input – the host computer software will require the user to run the test. Since it's not written yet, the exact required input is not available.

Failure Conditions – the test shall fail if any of the bad messages are parsed and broadcasted

Success Conditions –The test shall succeed if all of the messages are rejected

The price-tag unit's microcontroller is responsible for running the program described in figure 4.1.3. Its main role is to receive packets, make sure the packet is meant for that specific price-tag, and if it is it will parse the packet and use it to update the e-ink display.

The following is a simple test procedure. Like the transmitter unit's test, specific details cannot be provided at this time but the purpose of the test is painted well.

1. Program the price-tag unit and transmitter unit with the programs written by Precision Wireless.
2. Use the host computer to send a packet with a tag ID that does not match the price tags and confirm that the price-tag rejects it.
3. Use the host computer to send a packet with the proper tag ID and confirm that the e-ink display is updated accordingly.

User Input – the host computer software will require the user to run the test. Since it's not written yet, the exact required input is not available.

Failure Conditions – the test shall fail if any of the e-ink display is updated with information not meant for that specific price-tag.

Success Conditions – The test shall succeed if the e-ink display is updated accordingly when the tag ID in the packet matches that price-tag's ID.

Note: If the display is not ready to be tested log output will suffice since the purpose of this test isn't to test the e-ink software.

5.1.8 E-ink Display Testing

Since, the e-ink component is used only for displaying characters; the testing will ensure that all characters in the standard ASCII table [10] shall be displayed. Hence, the tester shall ensure all characters in the ASCII table can be displayed on the e-ink display.

Failure Conditions -The test shall fail if at least one character in the ASCII table cannot be displayed correctly.

Success Conditions -The test shall succeed if all the characters in the ASCII table are displayed.

5.2 System Integration

Once, the individual testing of each components namely the database, transceiver and e-ink display is completed, system integration would involve connecting the components as shown in the figure below. The Central Transmitter(Central Server in the figure) will interact with the database application through USB connection, while the LCD and the tag will communicate through SPI module.

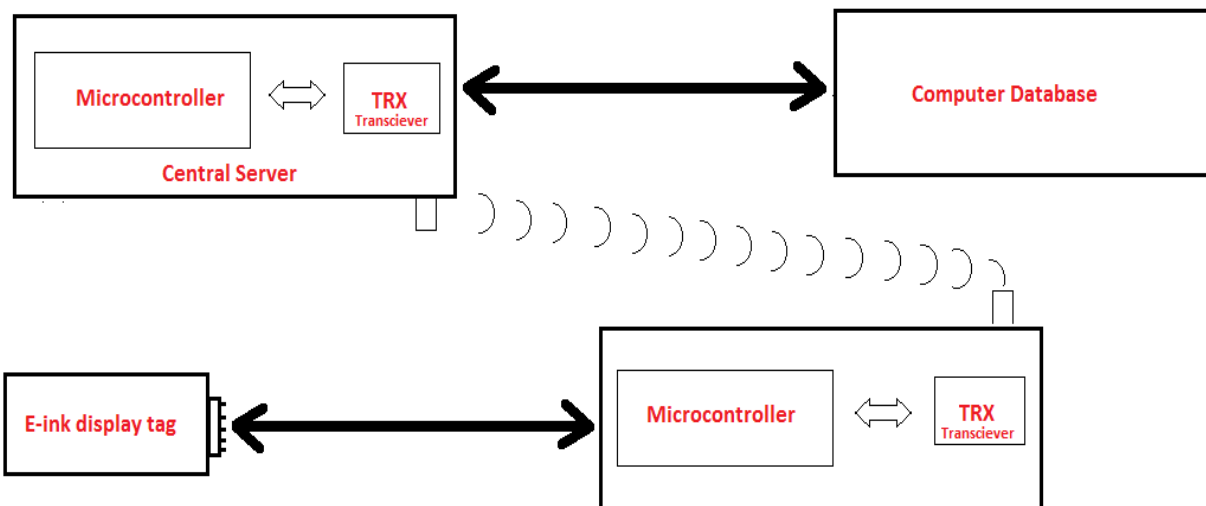


Figure 5.2.1:Figure illustrating the integrated AccuTag system

5.2.1 System Testing

System Testing for AccuTag version 2.0 would require at least 20 tags and a central transmitter unit with a fully functional and accessible database. System testing for the whole AccuTag system is described below.

1. The tester shall upload the standard price tag catalogue designed by Precision Wireless for 20 tags in the database.
2. The tester shall execute send instructions on the database Application thereby enabling the central transmitter unit to send information.
3. The tester shall ensure that all 20 tags have been updated with the information as specified in the price catalogue.

Failure Conditions -The test shall fail if at least one of the tags does not get updated or gets updated with the wrong information.

Success Conditions -The test shall succeed if all of the tags are updated with the correct information.

Sources and References

- [1] Atmel Corporation, "Atmel," 01 02 2012. [Online]. Available: <http://www.atmel.ca/Images/doc8359.pdf>. [Accessed 10 10 2013].
- [2] Atmel Corporation, "Atmel," 01 03 2013. [Online]. Available: http://www.atmel.ca/Images/Atmel-42081-RCB256RFR2-Hardware-User-Manual_Application-Note_AVR10004.pdf. [Accessed 10 10 2013].
- [3] Atmel Corporation AVR, "Atmel," 01 02 2013. [Online]. Available: http://www.atmel.com/Images/Atmel-8393-MCU_Wireless-ATmega256RFR2-ATmega128RFR2-ATmega64RFR2_Datasheet.pdf. [Accessed 10 10 2013].
- [4] Atmel Corporation Xplained, "Atmel," 01 05 2013. [Online]. Available: http://www.atmel.com/Images/Atmel-42079-ATMEGA256RFR2-Xplained-Pro_User-Guide.pdf. [Accessed 10 10 2013].
- [5] Pervasive Display, "Pervasive Display," 17 06 2013. [Online]. Available: http://www.pervasivedisplays.com/kits/ext_kit. [Accessed 10 11 2013].
[http://www.pervasivedisplays.com/products/27 "Documents"](http://www.pervasivedisplays.com/products/27_Documents) [Accessed 10 11 2013].
- [6] Atmel, "Atmel.ca," 25 02 2013. [Online]. Available: <http://www.atmel.ca/tools/ATMEGA256RFR2-XPRO.aspx?tab=documents>. [Accessed 10 10 2013].
- [7] Pervasive Display, "Pervasivedisplays," 26 06 2013. [Online]. Available: <http://www.pervasivedisplays.com/products/27>. [Accessed 10 11 2013].
- [8] D. L. Jones, "alternativezone," 10 06 2004. [Online]. Available: <http://www.alternatezone.com/electronics/files/PCBDesignTutorialRevA.pdf>. [Accessed 10 11 2013].
- [9] Texas Instruments, "Texas Instrument (TI)," 10 02 2011. [Online]. Available: <http://www.ti.com/lit/an/scaa048/scaa048.pdf>. [Accessed 14 11 2013].

Appendix A. Schematics

The figure A1 gives a detail schematic of the STB, which was obtained from the Atmel user guide and modified for use in the AccuTag Transmitter Interface System.

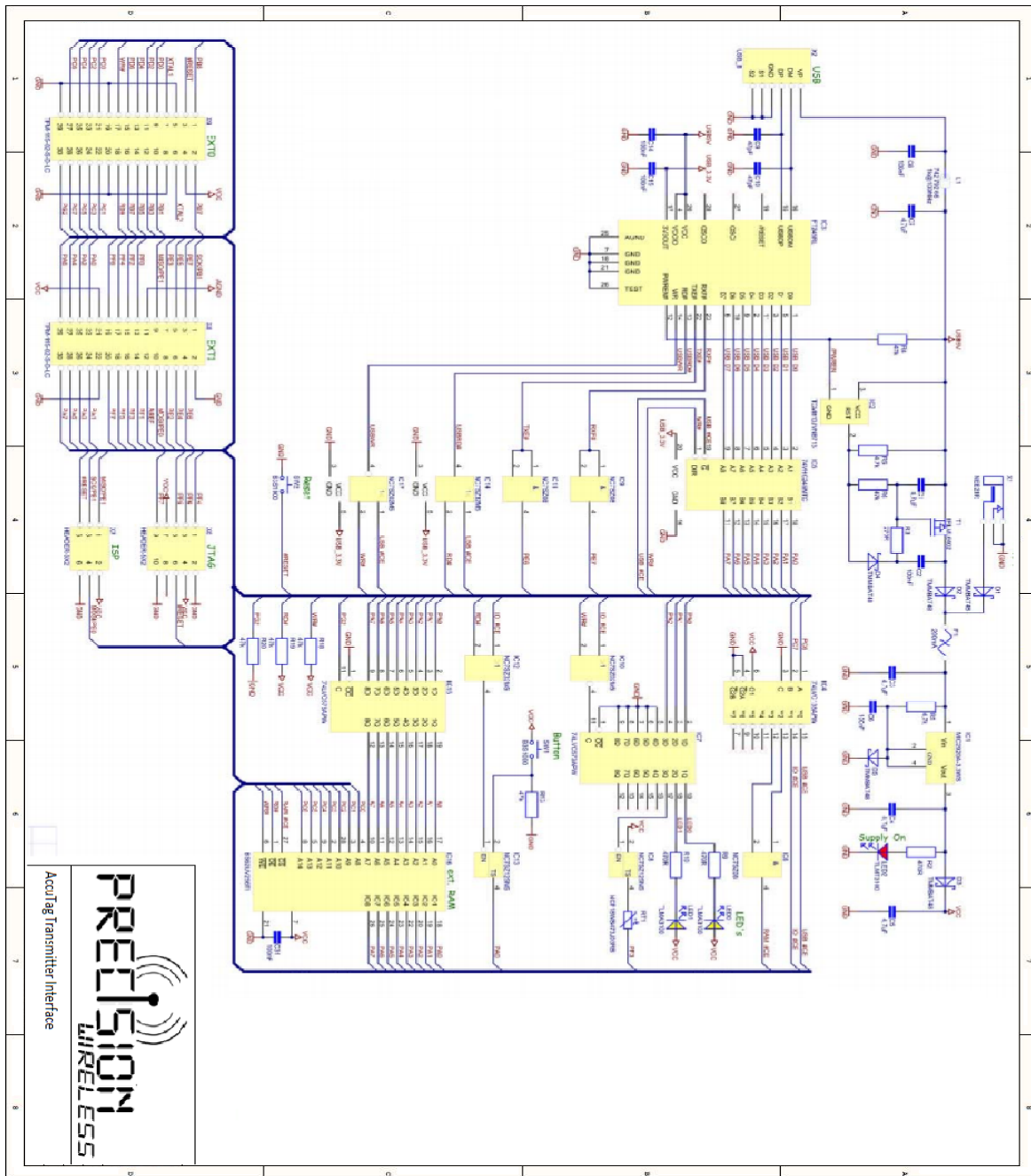


Figure A1: AccuTag Transmitter Interface (STB)

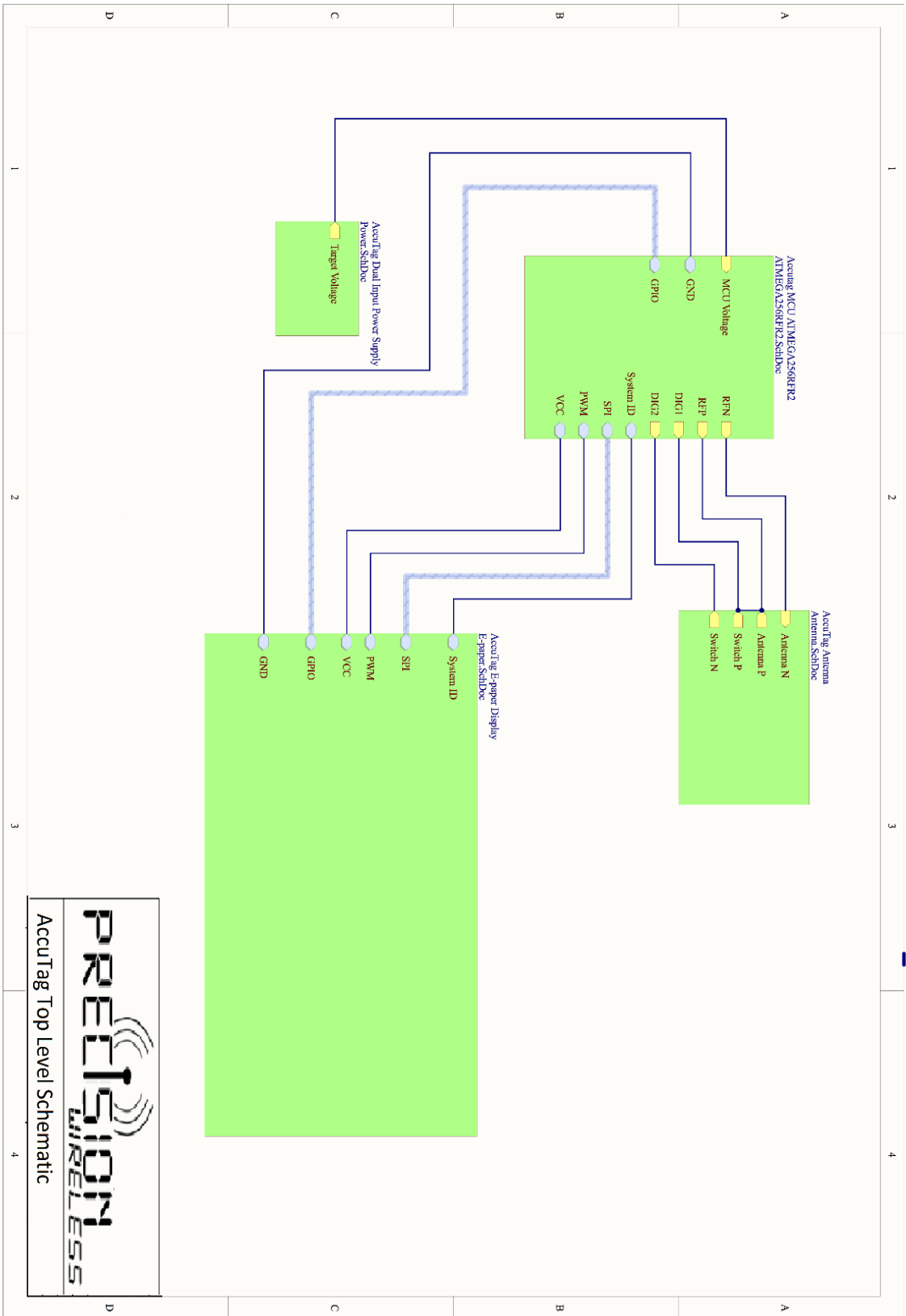


Figure A2: AccuTag Display Unit top-level hardware diagram

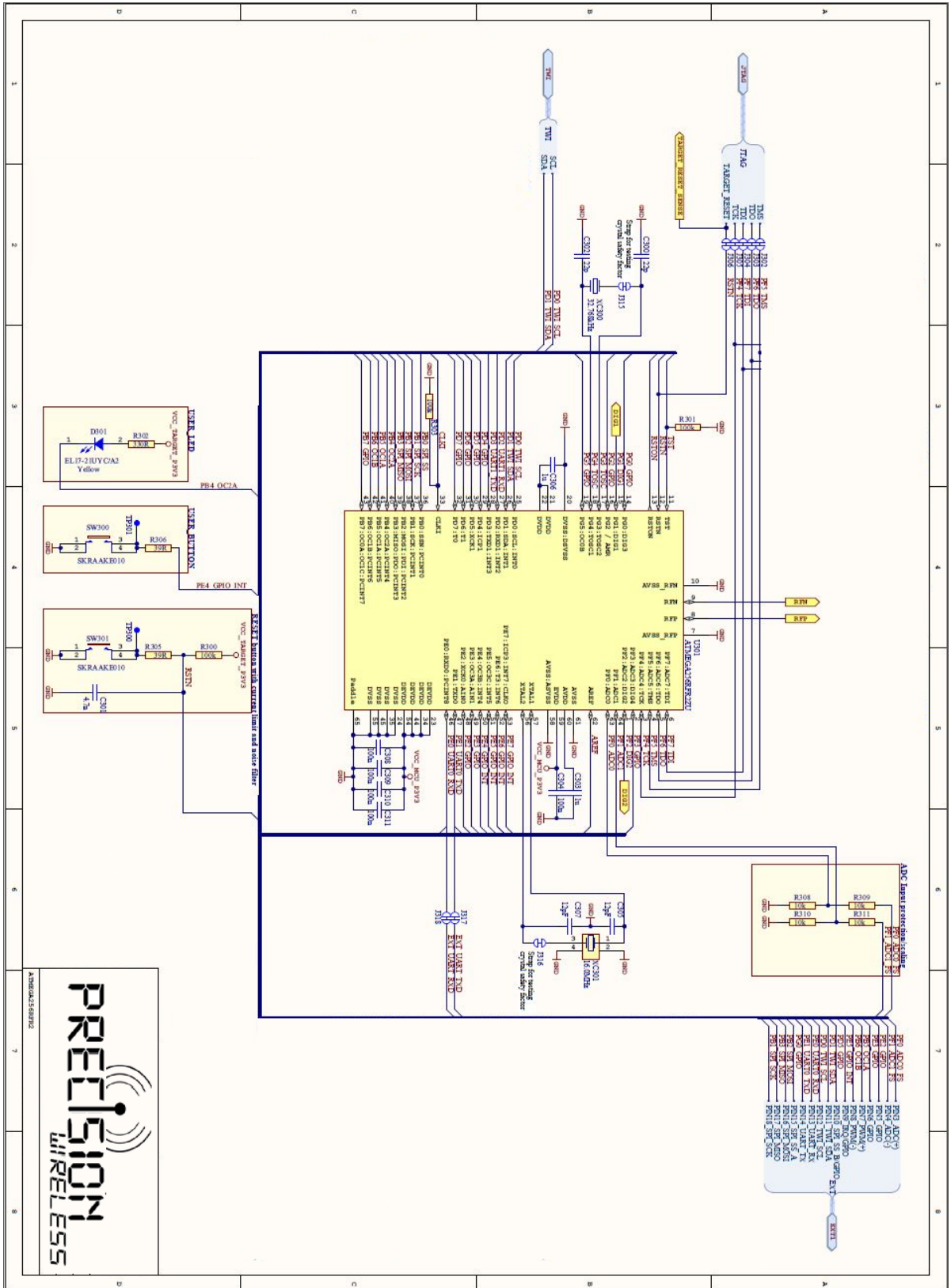


Figure A3: The AccuTag Display_ATmega256RFR2 connections to EXT1

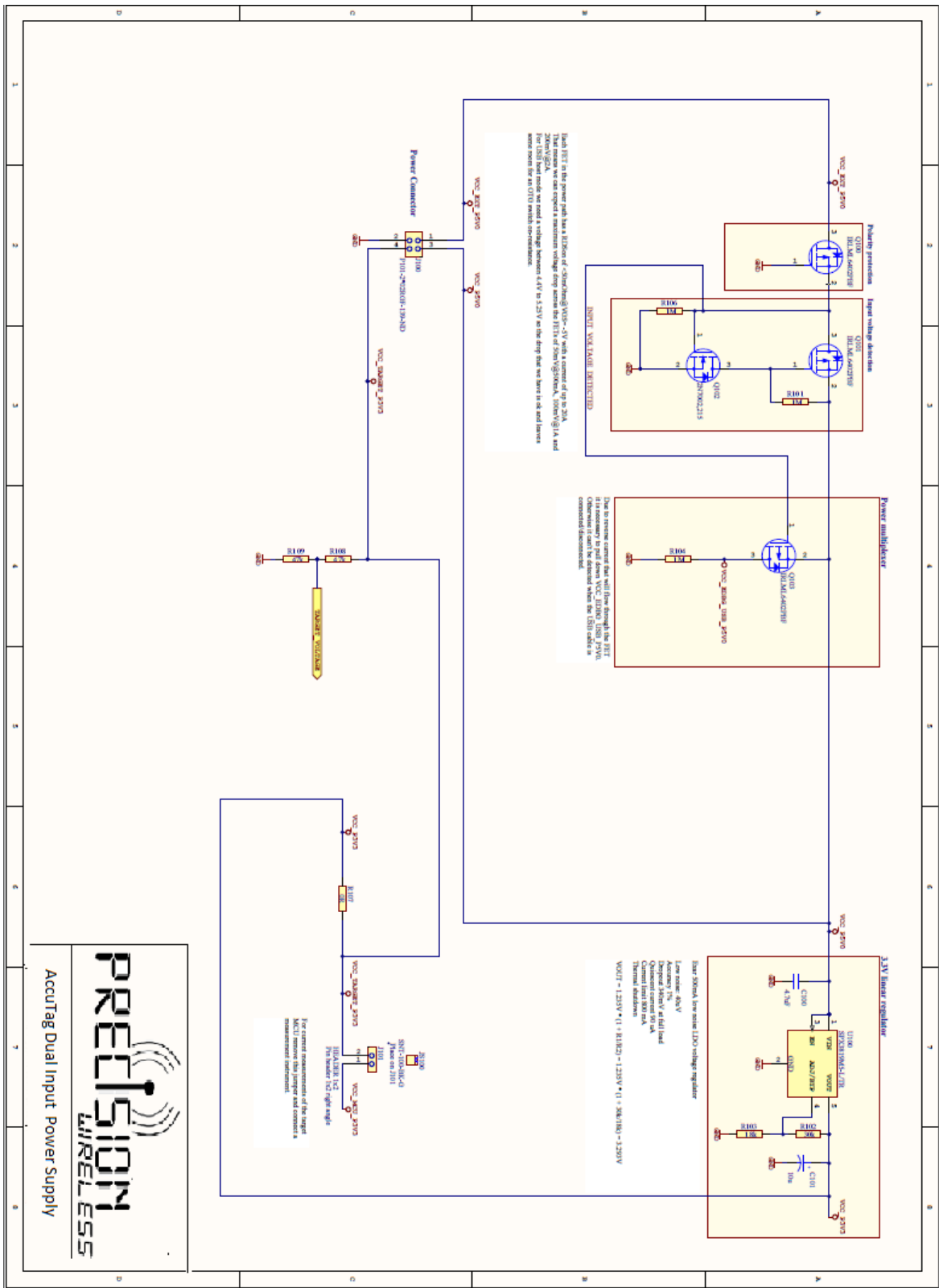


Figure A4: The AccuTag Display Dual Input power supply

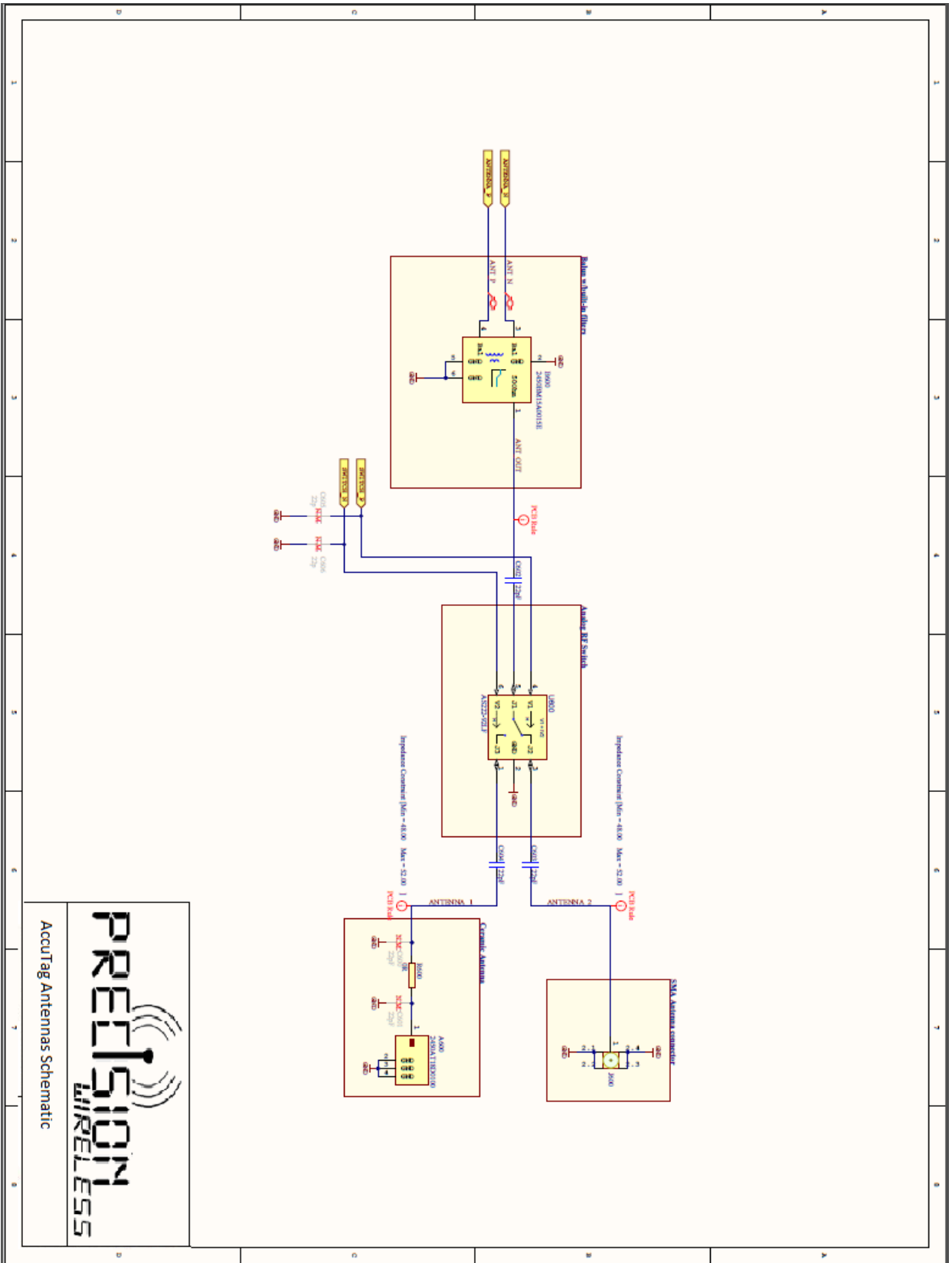
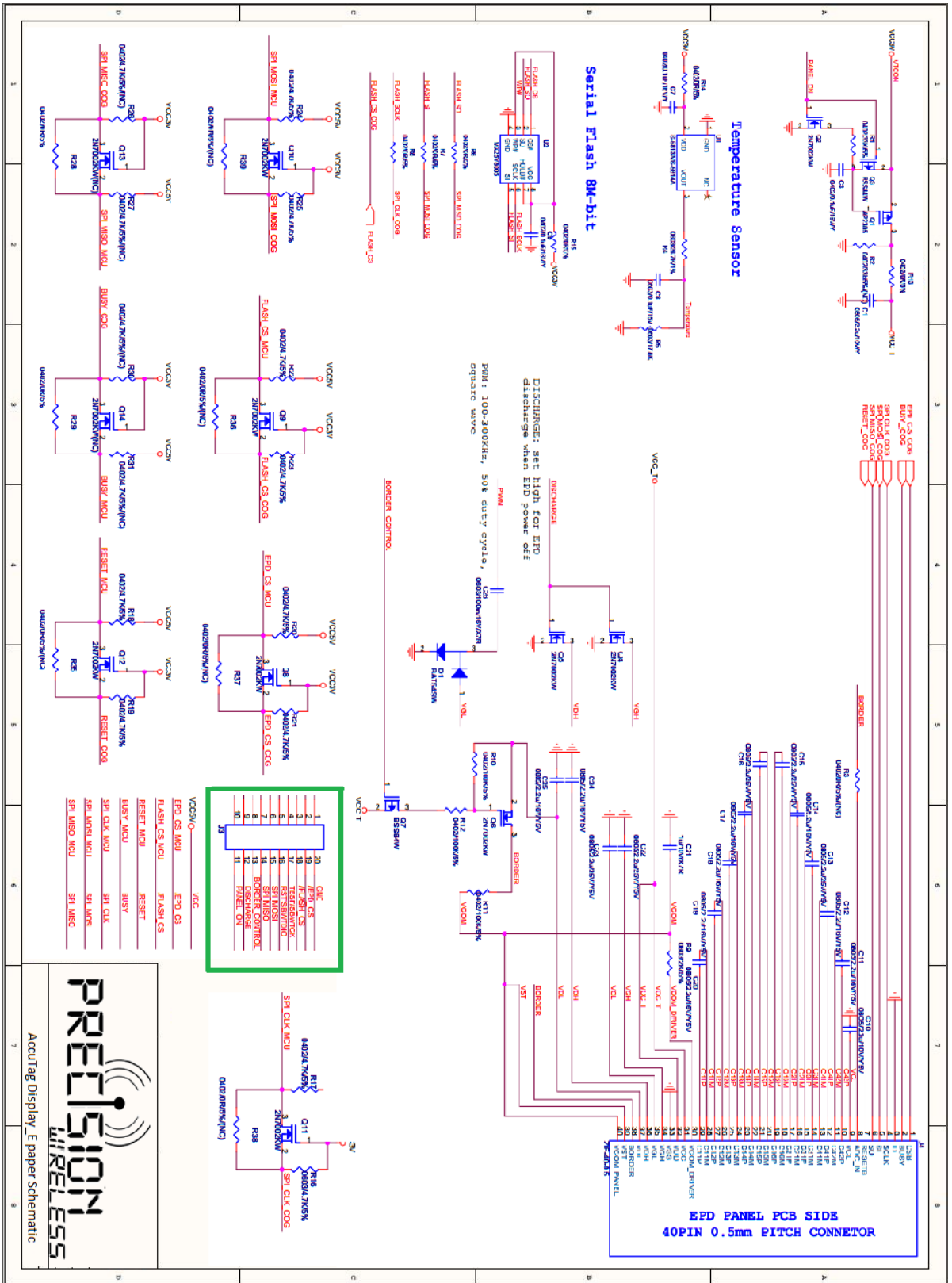


Figure A5: The AccuTagDisplay Unit Antenna

PRECISION WIRELESS
AccuTag Antennas Schematic



FigureA6:AccuTag Display part 1_E paper schematic

#	Designator	Quantity	Name/Value/Tol%	Description
1	R1	1	0402/33k/5%	Resistor
2	Q2	1	2N7002KW	Mos Transistor
3	Q3	1	BSS84W	Mos Transistor
4	Q1	1	AP2305	Mos Transistor
5	C3	1	0402/33k/5%	Capacitor
6	R2	1	0402/33k/5%	Resistor
7	R13	1	0402/0R/5%	Resistor
8	C1	1	0805/2.2u/10V/Y	Capacitor
9	R14	1	0402/0R/5%	Resistor
10	C7	1	0402/0.1uF/16V/Y	Capacitor
11	U1	1	S-5813A/S-5814A	Temperature Sensor chip
12	R4	1	0603/26.7K/1%	Resistor
13	C8	1	0603/0.1uF/16V	Capacitor
14	R5	1	0603/17.8K/1%	Resistor
15	U2	1	MX25V8005	Flash 8bit chip
16	R15	1	0402/0R/5%	Resistor
17	C9	1	0402/0.1uF/16V/Y	Capacitor
18	R6	1	0402/0R/5%	Resistor
19	R7	1	0402/0R/5%	Resistor
20	R8	1	0402/0R/5%	Resistor
21	U3	1	TPS77030	Voltage Follower chip
22	R33	1	0603/0R/5%	Resistor
23	C29	1	0603/0.1uF/Y5V	Capacitor
24	C30	1	0805/2.2uF/16V/Y5V	Capacitor
25	R34	1	0603/0R/5%	Resistor
26	C27	1	0603/0.1uF/Y5V	Capacitor
27	C28	1	0805/2.2uF/16V/Y5V	Capacitor
28	R32	1	0603/0R/5%	Resistor
29	R3	1	0402/0R/5%	Resistor
30	C10	1	0805/2.2uF/10V/Y5V	Capacitor
31	C11	1	0805/2.2uF/16V/Y5V	Capacitor
32	C12	1	0805/2.2uF/16V/Y5V	Capacitor
33	C13	1	0805/2.2uF/25V/Y5V	Capacitor
34	C14	1	0805/2.2uF/16V/Y5V	Capacitor
35	C15	1	0805/2.2uF/25V/Y5V	Capacitor
36	C16	1	0805/2.2uF/25V/Y5V	Capacitor
37	C17	1	0805/2.2uF/16V/Y5V	Capacitor

38	C18	1	0805/2.2uF/16V/Y5V	Capacitor
39	C19	1	0805/2.2uF/16V/Y5V	Capacitor
40	C20	1	0805/2.2uF/16V/Y5V	Capacitor
41	R9	1	0603/2K/5%	Resistor
42	C21	1	1uF/10V/X7R	Capacitor
43	C22	1	0805/2.2uF/25V/Y5V	Capacitor
44	C23	1	0805/2.2uF/25V/Y5V	Capacitor
45	Q4	1	2N7002KW	Mos Transistor
46	Q5	1	2N7002KW	Mos Transistor
47	C24	1	0805/2.2uF/16V/Y5V	Capacitor
48	C25	1	0805/2.2uF/16V/Y5V	Capacitor
49	Q6	1	2N7002KW	Mos Transistor
50	R10	1	0402/100K/5%	Resistor
51	R11	1	0402/100K/5%	Resistor
52	R12	1	0402/100K/5%	Resistor
53	Q7	1	BSS84W	Mos Transistor
54	C26	1	0603/100n/16V/X7R	Capacitor
55	D1	1	BAT54SW	Diode
56	D2	1	BAT54SW	Diode

Table A1: Bill of Materials