Modular PCB for John Deere Equipment

ECE445 Design Document

Sumanendra Sanyal, Sam Huhta, Zachary Hoberg
Team 17
TA: William Null
17 Mar 2020

1. Introduction

1.1 Problem and Solution Overview

Problem: John Deere currently manufactures an autonomous residential lawn mower, the Tango, that uses a buried wire to define the boundary of the yard. Installing this wire is a significant cost, requiring a site visit by specially-trained Deere technicians, and the related technology severely limits the mower's capabilities. For these reasons, John Deere is actively working to eliminate the boundary wire by using a localization algorithm fusing several sensors, such as stereo cameras, GNSS receivers, and ultrasonic sensors. However, the current Vehicle Control Unit (VCU) that operates the mower does not have the necessary processing power to run their localization and automation algorithms. Furthermore, John Deere would like to be able to apply their developed automation solution to other implements the company produces, but the current hardware is specific to the residential mower.

Solution: We have decided to replace the existing VCU with a modular design consisting of a main board, a perception board, and a vehicle board. The main board will handle the high-level autonomy universal to all equipment, the perception board will apply machine-learning algorithms to interpret the data provided by a specific set of sensors, and the vehicle board will handle all low-level algorithms specific to a given piece of equipment. For our project, we will design the vehicle board for the Tango mower. The Tango vehicle board will demonstrate the modularity of the system by communicating with the main board to run the low-level vehicle control software.

1.2 Visual Aid

1.3 High Level Requirements

- The 1-core processor on the vehicle board will boot a Linux operating system and run the provided algorithms to compute motor commands and mower state feedback at a minimum update rate of 20 Hz.

- The vehicle board will control the rotation speed and direction for all motors based on commands from the main board. The vehicle board will also charge the Tango mower's 25-volt lithium ion battery.

- The vehicle board will communicate with the main board via Ethernet through an off-the-shelf Ethernet switch, using standard Ethernet protocol, at a minimum update rate of 20 Hz. It will receive linear and angular velocity drive commands and blade motor commands from the main board, and it will provide feedback on vehicle odometry and blade motor state to the main board.

2. Design

2.1 Block Diagram

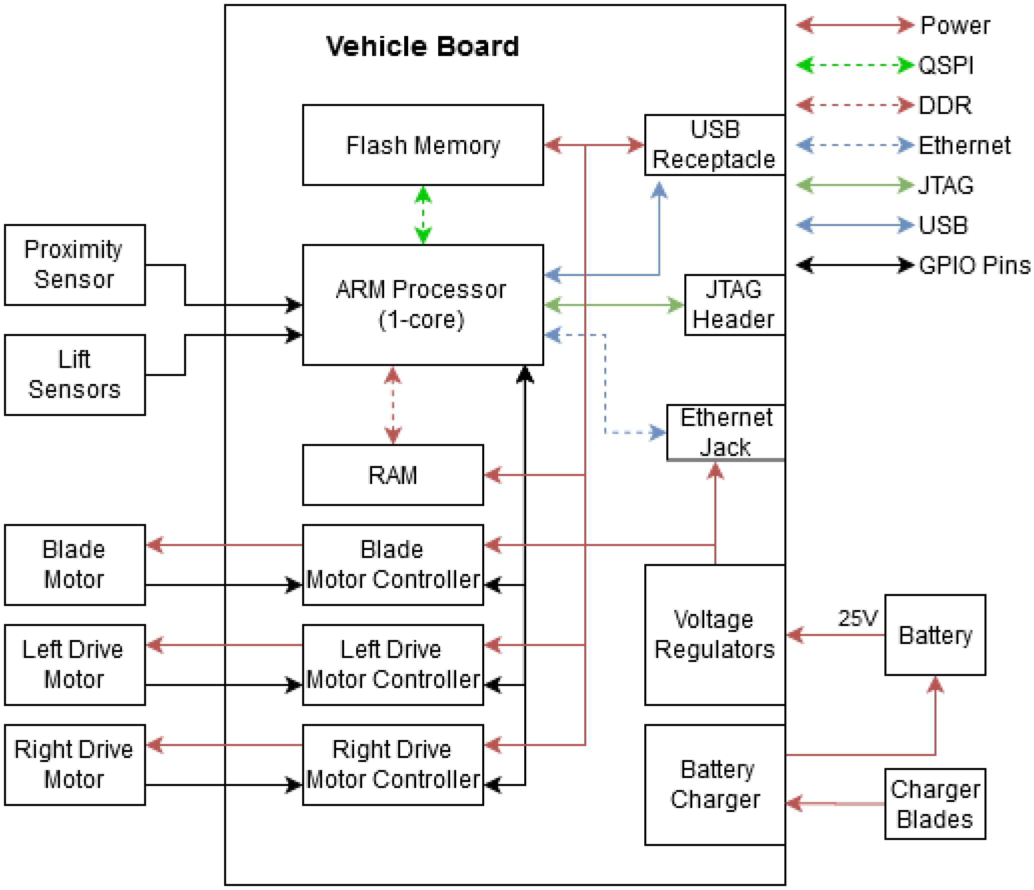

Figure 1. Modular control system Block Diagram

2.2 Vehicle Board Components

ARM Processor LS1012A

The LS1012A, RAM size, and FLASH memory size were chosen based on discussions with John Deere engineers regarding the sort of tasks/algorithms the vehicle board will need to be capable of.

Requirements	Verification
1. Processor boots Linux. 2. Processor can run vehicle board programs provided by John Deere.	1. Connect to the vehicle board via PuTTY terminal, run Linux terminal commands on the vehicle board. 2. Connect to the vehicle board via PuTTY terminal, and program the board to run a vehicle board program. Test the vehicle board response to commands over Ethernet cable by observing the response on the motor controllers. An oscilloscope or actual motor may be used for this purpose.

RAM

W632GU6NB09J

2 Gb (128Mx16) DDR3L SDRAM

The RAM consists of 4 parallel 2 Gb DDR3L chips in order to enable most efficient use of the 64-bit data bus width. This allows all of the 64-bit double values used to be stored and retrieved with maximum efficiency.

Requirements	Verification
1. Processor is able to store data to RAM and recall it properly.	1. Execute test code that stores values to RAM from processor and recalls them. NOTE: if Linux boots, RAM can be considered to be functional, as the Linux kernel code will be in the RAM.

FLASH

MX66U2G45GXRI00

Quad SPI NOR flash memory, 2 Gigabit

The LS1012A always boots from FLASH memory and the FLASH memory will contain the u-boot bootloader and Linux distro as detailed in the QorIQ Linux SDK.

Requirements	Verification
1. Processor is able to successfully boot from flash memory. 2. Flash memory is able to correctly store and retrieve values.	1. Connect to the vehicle board via PuTTY terminal, run Linux terminal commands on the vehicle board. 2. Connect to the vehicle board via putty terminal, and run scripts that store and retrieve values.

JTAG

The JTAG interface for an ARM processor can be used to take control of the processor directly with a JTAG programmer. NXP semiconductors has an expensive solution (~$6,000 from online quote) consisting of the CodeWarrior software (an IDE) and the CodeWarrior TAP probe which connects a computer to a JTAG pinout. To boot Linux, first the u-boot bootloader needs to be placed on the FLASH memory, which can only be done by taking control of the processor with a JTAG programmer.
Then the board can boot (via u-boot) and be connected to a host computer via an Ethernet cable.
Finally, via TFTP over Ethernet, the Linux distro can be installed on the flash.

The procedure for getting u-boot and Linux running on the LS1012A can be found in the CodeWarrior documentation([1]), the QorIQ SDK documentation([3]), and the LS1012A RDB documentation([2]).

QorIQ - family of ARM processors from NXP, of which we have chosen the LS1012A

The QorIQ Linux SDK(software development kit) contains u-boot and Linux kernel images that need to be deployed to the on board FLASH memory. The "System Recovery" section of its documentation details the procedures for using CodeWarrior to program u-boot on the FLASH memory. The "deployment" section contains several different methods for deploying the Linux kernel onto a reference board.

The LS1012A RDB(reference design board) is a reference board for this processor. The reference design board ships with u-boot on its FLASH. Our board will closely follow the design so the required connections for getting Linux up and running are present.

Requirements	Verification
1. JTAG can be used for debugging the processor. 2. JTAG can be used for installing u-boot. 3. Processor boots Linux and runs programs provided from John Deere.	1. Follow procedures for using JTAG for hardware debugging purposes by using CodeWarrior and following instructions detailed in its documentation. 2. Connect to processor via putty terminal, turn on the board, and look for the u-boot printout on the putty console. 3. Connect to processor via putty terminal, run Linux terminal commands on the processor. Run John Deere scripts on the vehicle board, assess how the board responds to commands from Ethernet.

Motor Controller Circuit

The motor and motor controller are already tested and running on the Tango mower VCU, and our vehicle board is replicating the circuitry but replacing the processor whose GPIO pins connect to the motor controller. In other words, this is part of our board but not a part of our design! The A3930 (motor controller IC) in combination with 6 power MOSFETS produces three PWM signals that drive a 3-phase motor. The 3-phase motor provides a Hall sensor feedback to the A3930 motor controller so the A3930 can adjust the PWM signals based on the speed command (from GPIO) and Hall sensor feedback.

Requirements	Verification
1. The motor controller circuit behaves identically to commands from GPIO pins on the VCU and on our vehicle board. 1a. Both the VCU and our vehicle board can execute commands to drive forward, back up, and turn left and right. 1b. Both the VCU and our vehicle board can provide feedback from the motors.	1. Execute test codes on the VCU and vehicle board that modify the motor speed. Observe the output of the A3930 with an oscilloscope. 1a. One of the test codes will consist of a sequence of drive instructions. 1b. One of the test codes will spin each motor at various speeds, printing the actual speed to the test code terminal.

Voltage Regulators

The voltage regulators must provide several voltages in the correct sequence to the LS1012A and power the motor controller chip.

Requirements	Verification
1. Supply power according to processor datasheet: a. 1.35V ± 67mV @ 2A b. 1.8V ± 90mV @ 2.5A c. 3.3V ± 165mV @ 8A d. 0.9V ± 30mV @ 16A 2. The voltage regulators must turn on in the correct order and with the correct relative timing per the processor datasheet.	1. Each voltage output will be hooked up to a resistive load and current and voltage monitored to ensure output stays within range 2. The voltage outputs will be hooked up to an oscilloscope to confirm that they turn on in the correct order.

USB/Ethernet/SDHC

Ethernet is a required functionality from John Deere, and we are adding USB because it can be used for the putty terminal connection. SD card interface isn't required, but can provide additional storage.

Requirements	Verification
1. Processor is able to utilize Ethernet as a communication interface. 2. Processor is able to use an SD card. 3. USB can be used by the processor as a communication interface.	1. Execute test codes for each communication interface. 2. Use putty terminal (ssh).

Battery Charging Circuit

The circuitry currently used to charge the battery of the Tango mower is provided by John Deere; we will simply add it to our vehicle board.

Requirements	Verification
1. Charging circuit operates identically on both current VCU and our vehicle board, with no increase in maximum charge time.	1. Charge a Tango mower using the current VCU, and then charge it again using our vehicle board. Ensure that each case uses up the same amount of battery power by subjecting the Tango to the same conditions prior to each charge, and time how long each charge cycle takes.

2.3 Tolerance Analysis

In our design for the main board, the requirements from John Deere included a four-core processor. This processor had a very high maximum power draw, which necessitated careful monitoring of the processor temperature to ensure it did not exceed the maximum allowable temperature. For the vehicle board design, only a single-core processor is required. Therefore, we are interested in what the difference would be in heat generation and dissipation, and if any consideration is necessary for the thermal characteristics of this board.

The LS1012 core has a typical power dissipation of 985 mW when core utilization is at 50%, and a maximum power draw of 1.72 W. In addition, I/O power draw from the peripherals listed in the block diagram add another 447.5 mW typical power dissipation. In total, the power dissipation of the processor will be anywhere between 1.4 W typical and 2.17 W maximum.

From the processor datasheet, the thermal resistance from the junction to ambient is 27.2 °C/W on a four-layer board with natural convection. Based on this value, we expect a temperature differential from the processor to ambient of between 38 °C and 60 °C. If we assume ambient temperature of around 28 °C, the board would reach at most 88 °C, well below the maximum operating temperature of 105 °C.

However, we also need to consider the actual use case of the board. For the Tango mower, the board will be located inside a sealed plastic case, where temperatures can start to approach 50 °C. In this case, the maximum processor temperature would be around 110 °C, and on an exceptionally warm day the temperature could be even higher. If we would like to use the processor in this condition, we can approximate the maximum core utilization to be around 90% with no safety margin.

If higher core utilization and/or a larger safety margin are required, forced convection over the processor and circuit board could be considered. The processor datasheet gives a thermal resistance from junction to ambient of 22 °C/W on a four-layer board with forced convection at 1 m/s. In this case, the power differential should range between 30 °C at typical usage and 49 °C at maximum usage. This would allow full core utilization with a small safety margin of 6 °C when the ambient temperature inside the mower is 50 °C, although a heat sink could be added to decrease the thermal resistance and lower the processor temperature further.

3. Project Differences

3.1 Overview

Originally, we focused on the main board, since it's the critical element of the entire modular design. However, due to the COVID-19 pandemic, we lost access to the equipment we needed for testing, so we had to shift gears. Because the main goal of this project is to replace the Tango's existing VCU with something more modular and more powerful, we chose to work on the vehicle board.

3.2 Analysis

The navigation patterns supported by the current VCU are classified by John Deere as "random" and "spiral," and the boundary wire must trace a single closed loop around the entire yard. The processor used by the current VCU is unable to run more complex navigation algorithms, and everything is on a single board. With a modular design, different tasks can be assigned to different boards: the main board can run the high-level code that controls the mower's movement, blade speed, and blade height based on sensor input; the perception board can handle the machine-learning algorithms needed to interpret the sensor input; and the vehicle board can convert the mower's movement and blade operation to the correct motor drive signals.

The main board requires a quad-core ARM processor, and in our original tolerance analysis, we determined that a heat sink was necessary to prevent that processor from overheating, with a fan being necessary for very heavy usage. The vehicle board can run on a single-core ARM processor, which doesn't need a heat sink for safe operation at full core utilization, so long as the entire board is cooled by a fan.

Whereas the current VCU has all of its processing power on one board, our modular design spreads processing power across multiple boards. In addition, using ARM processors that can boot Linux allows John Deere software engineers to write Linux applications for everything they want to do, which is much easier than writing machine-specific code every time.

The main board allows the Tango mower to use a wider variety of navigation patterns, and it can even use input from the perception board to adjust its navigation pattern to one that best fits a given property. It also allows the user to configure the mower from their phone, without having to touch the mower itself. However, the main board is designed to be universal, so it doesn't need to know much about the Tango beyond what it is; the vehicle board handles all of the details, from blade state to battery charging.

Both the main board and the vehicle board use a 64-bit architecture, so signals can be exchanged without reformatting or bit slicing. The main difference is the signal update rate: the vehicle board updates everything at 20 Hz, while the main board also runs algorithms with update rates of 10 Hz and 1 Hz. This includes signals exchanged between boards - the main board exchanges signals with the vehicle board at a 20-Hz update rate and with the perception board at a 10-Hz update rate.

In summary, our main goals for this project were to improve the modularity and capability of the Tango mower's hardware; we improved modularity by splitting the hardware into three boards, and we improved capability by using ARM processors that can boot Linux.

4. Cost and Schedule

4.1 Cost

Part	Cost (Prototype)
MX66U2G45GXRI00 **Quad SPI Flash (2Gb)**	$24.02
D2516ECMDXGJD-U **4-Gigabit DDR3L SDRAM**	4x $3.82
USB receptacle USB1110-30-A	3x $.53
Motor Controller A3930KJPTR-T	3x$6.39
Processor LS1012AXN7KKB	$27.78
MOSFET	$1.50 (total, estimate)
Ethernet jack A121540TR-ND	$0.84
SD card receptacle Molex#: 104031-0811	$1.95
MAX8655ETN+ **Buck Regulator 25A**	$12.89
TPS53353DQPR **Buck Regulator 20A**	$6.58
AOZ2262AQI-10 **Buck Regulator 10A**	$1.08
TPS62827DMQR **Buck Regulator 4A**	4x $1.87
Miscellaneous[1]	$50.00
Multi-layer PCB (4 layer? 6 layers?) BGA reflow	$123.45

1.Miscellanous includes the various discrete parts that we have not accounted for individually at this time such as resistors, inductors, and capacitors required by various chips, as well as headers and test points required for measuring and verifying various signals.

4.2 Schedule

Week of (Monday)	Suman’s tasks/goals	Zach’s tasks/goals	Sam’s tasks/goals
20 April (presentation on 4-23)	Work on block-level slides	Work on presentation slides for introduction/problem statement, solution overview, high-level requirements, and block diagram	1. Create presentation 2. Work on differences slide and block-level slides for processor and motor controllers; also work on conclusion a bit
27 April	Work on final report	Work on final report	1. Start working on the final report
4 May	Work on final report and teamwork eval	Work on final report and teamwork eval	1. Finish up and hand in the final report 2. Complete teamwork eval

5. Ethics and Safety

Lawnmowers have a certain level of inherent safety risk due to the spinning blades, and in order to mitigate these risks our design will still follow guidelines from the IEC 60335-2-107 standard regarding battery-powered lawnmowers. However, our project should not have any direct impact on public safety, as the boards we develop will be for internal John Deere testing and development only, and will go through several revisions by their engineers before any possible production runs. Despite this, our design will still ensure that existing safety features on the Tango mower continue to function as expected.

In accordance with IEEE Code of Ethics #7, “to seek, accept, and offer honest criticism of technical work,” we will continue to meet with John Deere engineers weekly to provide updates on our progress, ask questions, and seek feedback on our work.

In accordance with IEEE Code of Ethics #9, “to avoid injuring others … by false or malicious action,” we have signed an NDA (non-disclosure agreement) with John Deere regarding the proprietary information they have shared with us to develop our project.

6. Citations

[1]-“CodeWarrior Development Studio for QorIQ LS series for ARM v8 ISA,” *Creating, building bareboard project*. [Online]. Available: https://docs.nxp.com/bundle/GUID-F97DCA91-E4C7-475C-B314-D15C603BAA10/page/GUID-5A152069-64A2-43A9-A2F5-A1BCE9B02155.html. [Accessed: 17-Apr-2020].

[2]-“Layerscape LS1012A Reference Design Board,” *NXP*. [Online]. Available: https://www.nxp.com/design/qoriq-developer-resources/layerscape-ls1012a-reference-design-board:LS1012A-RDB. [Accessed: 17-Apr-2020].

[3]-“QorIQ SDK v2.0-1703 Documentation,” *Recover system using CodeWarrior Flash Programmer*. [Online]. Available: https://docs.nxp.com/bundle/GUID-39A0A446-70E5-4ED7-A580-E7508B61A5F1/page/GUID-94A3AC6E-2F0F-40B7-B4FE-C9819BA63A7C.html#GUID-6E44B664-2DFD-4695-9801-701AD8CB0B9D. [Accessed: 17-Apr-2020].

[4]-“QorIQ® Layerscape 1012A Low Power Communication Processor,” *NXP*. [Online]. Available: https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/layerscape-communication-process/qoriq-layerscape-1012a-low-power-communication-processor:LS1012A?tab=Documentation_Tab. [Accessed: 17-Apr-2020].

[5]-“IEEE Code of Ethics,” IEEE. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 23-Feb-2020].