Montview Liberty University Journal of Undergraduate Research

Volume 2 | Issue 1 Article 5

2016

A Cyber-Physical System

Andrew Davis
Liberty University, ardavis8@liberty.edu

Dustin Bowe
Liberty University, dlbowe2@liberty.edu

Josiah Nagel
jnagel4@liberty.edu

A Cyber-Physical System

Team 02

Andrew Davis

Dustin Bowe

Josiah Nagel

Liberty University

Abstract

The team was tasked with the creation of an autonomous cyber-physical system that could be continually developed as a post-capstone class by future STEM students and as a means to teach future engineering students. The strict definition of a cyber-physical system is a computation machine that networks with an embedded computer that performs a physical function. The autonomous aspect was achieved through two sonic sensors to monitor object distances in order to avoid walls and obstacles. The integrated system was based on the Intel Edison computation module. A primary goal for future addition is automation capabilities and machine learning applications.

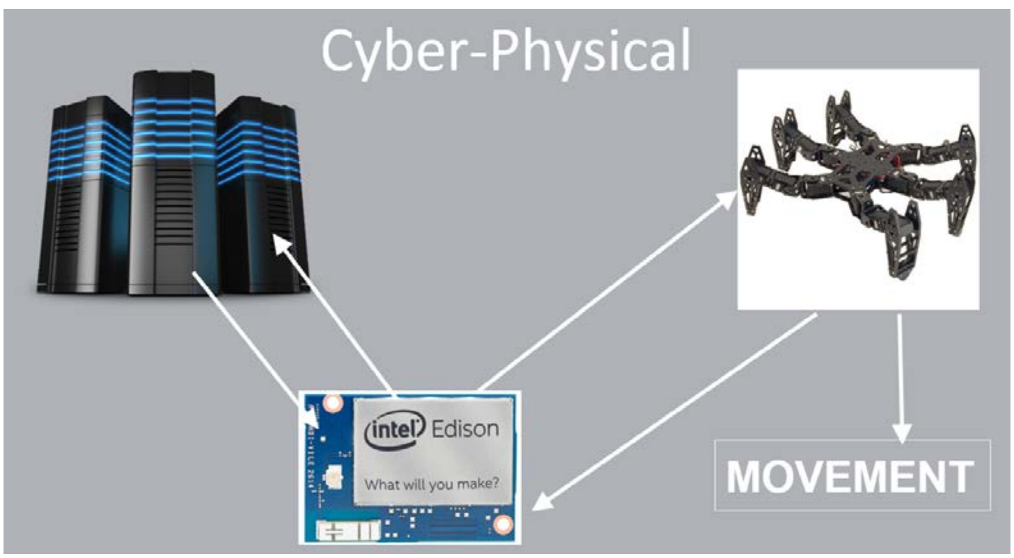
A Cyber-Physical System

The engineering department chair of Liberty University, Dr. Pettiford, requested capstone team 02 to create an autonomous cyber-physical system that could be continually developed to serve as a post-capstone class by future STEM students. The purpose of the system was to create a platform for future development that might impress future Liberty University engineering students with a cyber-physical system that would have a “wow” factor. The system was to be designed such that students could further develop the system by using their knowledge and skills in engineering and technology to add additional hardware, program additional commands, and improve the system, thus making the robot more advanced and complete. The final design was a hexapod, six legged robot that communicates with a base server to perform calculations and that would respond to the calculations by performing physical movement through no human interaction and fulfilling a cyber physical autonomous developmental system.

System Design Overview

The strict definition of a cyber-physical system is a computation machine that networks with an embedded computer that performs a physical function. The robot systems’ fundamental requirement is that it needs to be able to move from one point to another through a cyber-physical system setup. The system is generally described in the diagram below. Capstone Team 02 used a Liberty University Unix-based server as the computation machine. The embedded system is the Intel Edison along with two printed circuit boards (PCB) connected to 18 servos. The two PCBs are the sensor PCB and the Pulse Width Modulation (PWM) expansion board. The physical function performed is a coordinated movement of the 18 servos to create a walking movement in the robot.

The autonomous aspect is achieved through two sonic sensors to monitor object distances in order to avoid walls and obstacles. Automation harnesses sensors to receive data from its environment to perform functions without human control. The system contains a three degrees of freedom sensor, which is a 3D digital linear acceleration sensor, a 3D digital angular rate sensor, and a 3D digital magnetic sensor. These sensors are used to provide machine learning and artificial intelligence expansion on the robot for future student to add additional features. The machine learning capabilities can be augmented and added onto through the system's modularity. The robot is designed to be easily modified, and the PCB are placed to be easily accessible. The PWM board allows additional sensors, and the Intel Edison allows for multithreading. The multithreading can be used with the sonic sensors to avoid objects. The three degrees of freedom sensor gives positional information and can be used to collect robot position data. A GPS can be installed into the system which can consent autonomous positional movement.



Description of Functions

Server

The system was designed around a Wi-Fi router interaction from a laptop that mimics a server. The system can be controlled through a Liberty University server. The server performs complicated computations and system information storage. This capability lets the robot function by merely connecting to the server. Theoretically, an offsite location could be allowed to connect to the university server and control the robot from far off locations. Additionally the server allows the robot to use machine-learning methodology for complex path finding and environment processing.

Frame

A pre-built frame bought from Trossen produced a limited environment for developing the robot and accessing the internal circuitry. The system is a developmental platform so the circuitry needed to be more accessible than the original frame. The body of the frame that Liberty University purchased for team 02 was redesigned to allow access to all the circuitry without having to remove the bolts or dismantle the frame. The open frame allows easier access to the integrated system and increases room for additional hardware.

Integrated System

The integrated system is based on the Intel Edison. The Edison was chosen for its built-in Wi-Fi, physical size, CPU, and Linux operating system (OS). The Wi-Fi facilitates the robot's capacity to easily connect to the server for heavy calculations that he on board processor cannot handle. This also is how the robot will get its respective commands for its autonomous movement and directions for movement. The Edison's size is small enough to provide room for additional circuitry in the robot's frame. The system's central processing unit (CPU) has the ability to multithread up to four different threads. This gives one for server communication: two for sensor processing and one for movement processing. The Edison can have multiple operating systems. The Linux can communicate with the Linux based server easily as well as use the GCC to program with C++. The Linux can support the MRAA library which contains controls for all the Edison functionality including inter-integrated circuit (I2C), Pulse Width Modulation (PWM), and Universals Asynchronous Receiver Transmit (UART).

Code and Library

The existing system was replaced with a Linux OS and GCC which is essential in using C++ programming. MRAA was used as the C++ control library. The MRAA library allows for future expansion of the system since it has UART, I2C, SPI, and PWM control architectures. The I2C and SPI allows for sensor communication with sensors such as gyroscopes and accelerometers to name a few. The 3D system’s angular sensors receive a clock signal and input and output from I2C protocols. Each sensor data is tuned to the same clock and receives power through GPIO inputs as well as 3.3V. The output is determined by a variance in the return signal, which has a constant voltage output measure during no activity and during a steady state. Upon sensor movement, the steady-state level of voltage varies. The change in magnitude correlates with a sensor movement. The variance of the magnitudes’ unit is dictated by the datasheet of the product. This data will portray the change in elevation and tilt, acceleration, rotation speed, and rotation direction. UART is used to send control packets to the half-duplex servos. The packet format is

Where the packet section 0XFF alerts the servos to an incoming signal, the ID section is the servo number; the instruction section is the command to be performed; the parameter section is used when instruction requires ancillary data; and the check sum section lets the system know when the packet ends.

The code to control the servos is an application of Inverse Kinematics (IK). Inverse kinematics is a system of kinematic equations that have current positions for a robot and use joints as references. IK then receives a desired end position and reverse engineers a movement pattern for the servos to perform in order to reach the desired end joint position. The robot defines a set default position in flash memory. The position is 512, which is a right angle. On startup, the robot will set each servo to a default. From here, the system is set up to receive commands from a controller and calculate, using IK, an end position.

The system code is on the Ubilinux OS on the Intel Edison. This couples well with WI-FI and allows the previously mentioned GCC system. Servers are mainly Linux systems and interact well with the Edison.

Communication

The complete communication system creates links to every computer system. The robot links from the Liberty University server, through WFI, to the Intel Edison. The Intel Edison links to the servos and sensors. The sensors and servos link to the Edison, and from the Edison to the server.

System Modularity

One focus of the project was to make the robot a developmental platform. This meant setting up the system to be easily modified by future students. Now the system code that interacts on the server does not restrict access to only that server. The team used a router connected to a computer as the team's private server during the development of the robot. This could be done equally for any server. If necessary, the system can be controlled by a human through a controller or Wi-Fi.

The Edison has been modified to allow additional PWM connections since sensors primarily run through PWM protocols. The current system has a setup where the sonic sensor connects to a server to allow the sonic sensor to spin. This servo is controlled with PWM protocols. Almost any generic sensor can be added to this robot system because the frame is an open design that allows easy circuit board access. Also, the battery was located above the circuitry and blocked access to the system components. The battery was moved to underneath the robot to remove the restriction

The system code is capable of being augmented and is well commented for future students to modify. The Linux system does not hard code the program so that each sensor added to the robot code can be written for that specific sensor.

Future Additions

A large goal for future addition is automation capabilities and machine learning applications. Automation harnesses sensors to receive data from its environment to perform functions without human control. A Linux webcam would allow the system to capture frame during motion to compare object size variance. This variance could easily be used to sense object depth and allow intricate autonomous movement through unmapped areas. Through machine learning, the robot can use the gathered data to recognize pattern and learn from its environment to make better route predictions.

Conclusion

Overall, the system without any sensors is a cyber-physical system. The addition of sensors gives the platform a large amount of capability and room for more improvements by future students. The robot can receive commands wirelessly from the server and interact with its environment autonomously.