ETD MINI-GRANT Final Report

Turning a non-collaborative robot into collaborative robot: an experimental step towards meeting industry 4.0 requirements

Project Director: Hadi Alasti

Assistant Professor of Electrical Engineering Technology

Phone: 260-481-6423

Fax: 260-481-5734

Email: alastih@pfw.edu

INSTITUTION: Purdue University Fort Wayne

School of Polytechnic

2101 E Coliseum Blvd., Fort Wayne, IN 46825

Preparation date: 09/30/2020

Edit date: 10/06/2020

Abstract

This report presents an experimental adaptation of a non-collaborative robot arm to collaborate with the environment, as one step towards adapting legacy robotic machinery to fit in industry 4.0 requirements. A cloud-based internet of things (CIoT) service is employed to connect, supervise and control a robotic arm's motion using the added wireless sensing devices to the environment. A programmable automation controller (PAC) unit, connected to the robot arm receives the most recent changes in environment and updates the motion of the robot arm. The experimental results show that the proposed non-expensive service is tractable and adaptable to higher level for machine to machine collaboration. The approach of this report has industrial and educational applications.

The ASEE ETD mini-grant's fund was spent to purchase materials, parts and units, such as communication related parts, sensors and PAC with industrial PC capabilities. The PAC, which is used in this project has been specifically designed for industrial internet of things (IIoT) application.

1. Introduction:

With the extension of industry 4.0 in manufacturing industries, the advanced manufacturing and automation machines are communicating to the other cooperating machines and also with the environment by using the embedded sensors in the machines' built. However, the existence of a large number of expensive programmable legacy automation machines in the manufacturing sites that are still in good working condition makes it difficult to add the industry 4.0 requirements in many of this site. Production adaptability, high efficiency, low environmental risk, less expensive final products, etc. are among the advantages of industry 4.0 over its legacy. Adapting the legacy machines to create similar environment such as industry 4.0 is the dream that this project tries experimentally investigate.

Cloud communication services has been one of the growing technologies in the past recent years. Data repository, online computations, access to data visualization services such as maps or graphs, and data sharing with convincing security level are just among the cloud-based services. The ever-more increasing number of cloud service providers and cloud brokers is a good evidence of market influence of this business.

Since a few past years, a large number of machines, homes, cars, etc. are connected to the Internet with this objective to make the world more connected. Internet of things (IoT) is a good mean for this connectivity. Many of the IoT services are provided over the cloud, which is also called cloud IoT (CIoT). CIoT services are getting cheaper, faster to start, and more convenient year after year. These services can provide the required machine connectivity, painlessly. In this project, instead of purchasing a domain and building IoT by adding online database, visualization services for data presentation, etc. we use CIoT services to connect a robot arm to its environment.

Variety of sensing possibilities, such as proximity sensing, touch, motion sensing, etc. are just among the options to include in this project, however due to the limited access during the pandemic situation of COVID-19, just a few of these sensors were embedded in the environment to let the robot arm collaborate with the environment, as example of making a safer manufacturing work environment.

As the implementation scenario, the proximity sensor reads the distance to the approaching body and sends this data to the CIoT, wirelessly. Robot controller, which is now connected to Internet through the PAC, checks for the distance information and setup the speed of the moving robot arm proportional with the distance of the approaching object, for safer environment. In implementation of this attractive scenario, embedding wireless sensors in the environment, C and Python programming languages and working with PAC/iPC that is directly attached to the robot arm, were experimented.

This multidisciplinary, attractive project covers topics in Electrical Engineering Technology, Computer Engineering Technology and Information Technology. By embedding industrial internet of things (IIoT), which is one of the main requirements of industry 4.0, the project introduces a major challenge for the students who like to be involved in solving a real-life problem in industry environment. This project that was implemented in several phases, has had a little bit trajectory modification in comparison to the project proposal, to match the chores with the available fund. One objective of this project is to involve the students in its outcomes in all of its focus fields..

2. Implementation of the project in big picture:

The project was implemented in four separate phases that will be discussed in next section. However, in big picture, it can be presented according to Figure 1.

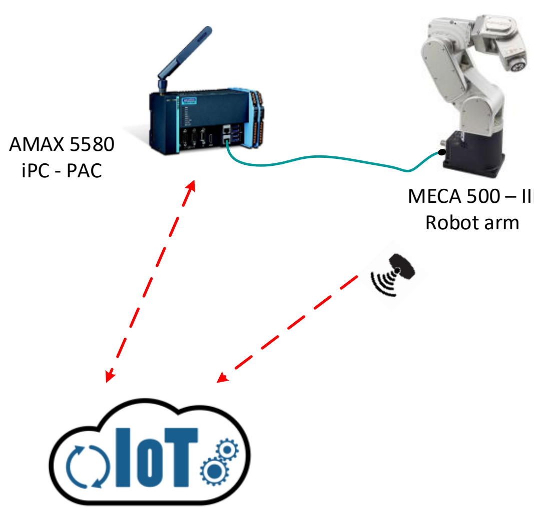

Figure 1: The implementation of the project in big picture. To make the robot arm to collaborate with the environment to make a safer work area, the proximity wireless sensors send the distance of the moving objects to the cloud. The most recent data is read by an industrial computer, where it controls the speed of the robot arm.

As Figure 1 illustrates, wireless sensing performs the detection of the events and its quantity (if applicable). Then detection result is projected to Internet and over the cloud. A program over the local robotic site is continuously checking for the new updates of the cloud data, and then it controls the behavior of the robot accordingly.

The study and implementation of this project was done in four phases that in the rest of this report the outcomes of each phase will be discussed.

3. Implementation phases of the project:

Phase 1: *Sensor and wireless communication devices*

Coordination with the environment and the other working machines demands to have sensing and communication with the work ecosystem. The sensors detect and measure the quantity of effect and the communication device, which is usually wireless, transfers the detection's outcome for storage over the cloud.

Variety of sensors such as proximity sensors (ultrasonic and optical), touch sensors, distance sensors, motion sensors, etc. were considered in this project. Among all, based on the access limitations due to the pandemic situation, ultrasonic proximity sensor with distance measurement capability was considered as one possible sensing option.

Besides the ultrasonic sensor, several other types of sensors such as ledar, vision sensor, etc. were purchased for future outgoing research paper, or for teaching purpose out of the granted fund.

For wireless communication purpose, several forms of wireless technologies were considered, such as WiFi, Bluetooth, Long Range Radio (LoRa) and Sigfox for low data rate transmission, and Long-Term Evolution-M (LTE-M) that is offered for machine to machine communication. Among these technologies WiFi technology with its good supported data rate and availability option was selected for the mentioned experiment of this report. However, to test the performance of the other technologies, devices with the other wireless communication capability were purchased. The performance of these technologies will be presented in future research work.

The software defined attribute of the wireless sensor platforms is just one of the advantages of these platforms that allowed this project to be implemented with flexibility. This feature made it possible to use variety of attached sensors to the software defined platform for optimal or sub-optimal detectability.

The list of the purchased wireless sensors and communication platforms is available in Appendix A of this report.

Phase 2: *Webpage / server setup and data collection*

In implementation of the Internet of Things (IoT) service, several researches were done in parallel to the phase 1. The requirements of the project, such as access to a web-space, preparing web-site over the leased web-space in accordance to the regulations of the web-host and installation and management of database and data visualization for recording and tracking the collected sensor data. Besides implementing all of the aforementioned needs to start an IoT project, using IoT service that are available over the cloud was the other possible solution that was eventually cheaper, faster and more convenient.

For this purpose, several researches were conducted to find the potential cloud-based IoT (CIoT) service providers. Table 1, presents a number of these service providers and the feature of their service. After this research, Thingspeak was selected due to the ease of work with its service, and the potential to use MATLAB routines for data analysis.

Table 1: List of potential Cloud-based Internet of Things (CIoT) service providers

Comparison of IoT Platforms					
IoT Software Platform	**Vender**	**Security**	**Protocols for data collection**	**Data Visualization**	**Database**
ThingSpeak	Mathworks	Basic Authentication	MQTT and HTTP	Yes	---
Google Cloud IoT	Google	Asymmetric key authentication over TLS 1.2	MQTT and HTTP	Yes	MySQL, PostgreSQL, SQL Server, and HBase to other traditional databases
IBM Watson IoT	IBM	FIPS 140-2 Level 4	MQTT and HTTP	Yes	--
AWS IoT Analytics	AWS	FIPS 140-2 Level 3	MQTT and HTTP	Yes	Ad hoc or Scheduled SQL
Countly	Countly	GDPR, HIPAA and COPPA	HTTP	Yes	MongoDB
ThingBoard	ThingBoard	Basic Authentication	MQTT, CoAP, and HTTP	No	Cassandra
Thing+	Daliworks Inc	Basic Authentication	MQTT	Yes	--
AT&T IoT Platform	AT&T	Basic Authentication	HTTP	Yes	--
SensorCloud	Lord	DoD (SRG) Impact Levels 2 and 4 and 5, DFARS	HTTP	Yes	FDBS
Axonize	Axonize	Basic Authentication	MQTT and HTTP	Yes	--
Cumulocity IoT	Software AG	ISO 27001, DSS and other standards	HTTP	Yes	MySQL,
Ubidots	Ubidots	X-Auth-Token	HTTP, MQTT and TCP/UDP	Yes	--
Apache iota	Apache	Basic Authentication	HTTP	No	--
Murano	Exosite	ISO 27001, DSS and other standards	HTTP	Yes	--
GroveStreams	GroveStreams	Basic Authentication	HTTP	Yes	--
Knowi	Knowi	Basic Authentication	MQTT and HTTP	Yes	MySQL, SQL Server, MongoDB

Phase 3: *IoT adapted programming to control the robot arm*

The next step on this project was finding a programming language that has the following two features:

- Be able to simply read data from the CIoT website.
- Be able to communicate with the robot arm as the legacy, programmable machine.

After conducting several researches and having several programming languages, it was realized that it is required first, to find a proper controller that may communicate with the robot arm using the specific programming language. Accordingly, the duty of this phase was mixed with that of phase 4.

Programmable logic controllers (PLC) with their ease of programming, industrial grade and their low price are favorite systems blocks that are seen almost everywhere in manufacturing and industrial sites. However, they are not necessarily the best option in controlling the programmable machines, due to their need to a gateway, as interpreters between their own language and the network protocol to communication with the robot arm. Accordingly, instead of stopping in PLC level, we promoted the search and decided to use a programmable automation controller (PAC). PACs with their communication capability and supporting variety of network protocols can communicate with machines with less limitations.

Against this simplicity, the higher expense of buying one PAC was the other major limitation. After several researches, and reading the features of the PAC units of several automation controller manufacturers, it was decided to buy one base-unit of AMAX 5580 at the cost of nearly $1500.

AMAX 5580, is an industrial personal computer (iPAC) – PAC that supports several communication protocols and ports such as Gigabit Ethernet, RS-232 and RS-485, USB-3, HDMI. The device is modular and a greater number of aforementioned ports can be added to the base unit. Also, the iPC-PAC allows to add EtherCAT modules to directly communicate with industrial units. The unit may work using LabView and other Microsoft Visual Studio software pieces.

After selecting the AMAX-5580, in the short list of programming languages, Python was selected, due to the following features:

- Python is a script language and it is relatively easy to work with.
- Python is available for free, so based on the limited available fund, it was worriless.
- Working with Python in network programming to collect data from Web and programming the robot arm based on a network protocol (EtherCat) is possible.
- Python is the most loved programming language and it is worthy to invest for it!

After selecting the programming language and the APC, it was time to implement the code and start Phase four. Of course, even before this step the programming of the wireless sensor platforms in working with the CIoT websites was started, however the serious step of this project was involvement of the legacy robot arm with the PAC and the selected programming language.

Phase 4: *IP-based ~~PLC~~ PAC-controlled robot arm*

The available robot arm, MECA-500 III, a mini robot arm with EtherCAT industrial protocol for communication is connected to the purchased AMAX-5580 iPC via Ethernet port as master controller port. To control the robot arm, a static IP address was assigned to the module on Ethernet network interface card (NIC). The AMAX-5580 unit is connected to internet via wireless network interface card (WNIC) to check out the CIoT data, based on the Python code that runs over the AMAX-5580 module. Unlike NIC, DHCP was assigned for the WNIC for dynamic IP addressing that is given by the wireless router in the work area.

The Advantech AMAX-5580 was ordered after mid-August and it was delivered on September the 24th. The reason for the late delivery was that this PAC is pretty new and less than 100 units of this brand has been sold, worldwide. On the shopping day, only 19 units of this PAC were sold in the United States.

Figure 2.a, illustrates the purchased unit, which is the base-unit for AMAX-5580 PAC. The base-unit can accept multiple EtherCat and other standard port modules to simultaneously handle several automation units. Figure 2.b, presents a AMAX-5580 with two additional EtherCat modules in its right side (courtesy of Advantech.com). AMAX-5580 has been specially designed for industrial Internet of Things (IIoT) applications.

(a) Connected AMAX-5580 unit in experiment

(b) AMAX-5580 with EtherCat modules (Courtesy of Advantech)

Figure 2. Advantech AMAX-5580, the iPC/PAC, is modular and its base-unit has multiple communication ports such as GbE, WiFi , USB-3, RS232 and RS485. Figure 2.a, shows the purchased unit connected to the MECA 500 robot arm. Figure 2.b, shows the added EtherCat units to the iPC/PAC.

4. Scenario: Implementation of an adaptable invisible fence:

As one application instance of using CIoT, in this section the preparation of a non-collaborative robot arm to the environment to collaborate with, is presented. Here the augmentation of the wireless sensors to the environment in order to modify its motion speed proportional with the distance of the observed object is discussed. In this implementation, the robot arm reduces its speed as a moving object approaches its location. The speed is proportional with the distance to the location of the sensor, until the robot may stop.

For this implementation, one ultrasonic sensor, which is directly attached to one ESP8266 Arduino (also known as NodeMCU), senses the distance to the closest object. ESP8266 has one WiFi channel that is compatible with IEEE 802.11 b/g/n. The proximity sensor's report to the ESP8266 microcontroller is analyzed in the microcontroller and then the distance to the closest object is calculated and it is sent to the IoT site, over the cloud via WiFi. In this project we used Thingspeak IoT service. At first, the free IoT service of this site was used and then, I purchased an academic account due to the long minimum period of sensor report of the free account (15 second) to the CIoT site. The academic account, at the cost of $250 per year reduces the minimum period of sensor report to 1 seconds.

The circuit in the bottom left side of Figure 3, which is build on white breadboard is the wireless sensor assembly. ESP8266 is connected to a local WiFi that is formed by one access point (AP) in its neighborhood. The sensor reports are sent to one Thingspeak channel that is protected by the "access key". The instantaneous sensing results are projected on a graph on Thingspeak.

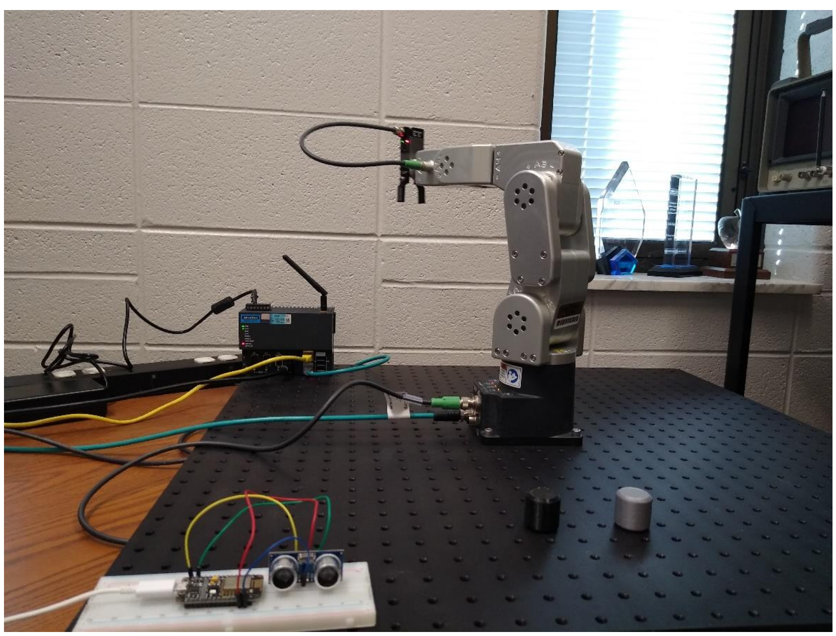

Figure 3. The connected MECA 500 robot arm to AMAX-5580 iPC/PAC in an experimental test to implement an invisible fence using ultrasonic proximity sensor.

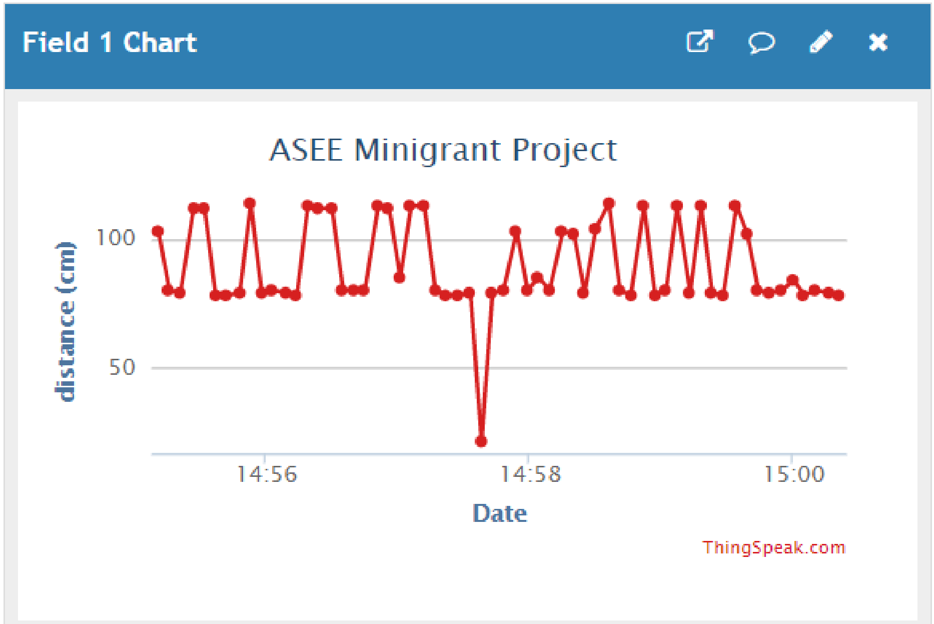

Figure 4. The collected data on Thingspeak, CIoT service

Figure 4, illustrates the sensing record of the distance to the ultrasonic sensor in cm. These data recorded after using an academic account.

On AMAX-5580, we prepared a Python code for two purposes. First, it continuously checks the Thingspeak site for the assigned channel and with the related access key, and gets the most recent changes to the proximity sensor. Second it controls the 6-axis robot arm's motion and its speed. Once the reported distance is short, it proportionally drops its speed in order to avoid collision. It is required to mention that the AMAX-5580 is connected to a monitor via one HDMI cable and the Python code is compiled and run for the robot arm on AMAX-5580.

The robot arm is connected to AMAX-5580 via one Ethernet cable (the green cable in Figure 3) with RJ45 jack in the master side. The power to the robot arm is provided by a 24 V, special DC power supply via the black cable in Figure 3, that is connected to the base of the robot arm.

A short video record of the reaction of the robot arm to the distance with the free Thinkspeak account (15 seconds update period) is uploaded on Youtube for illustration of the reaction of the robot to the change in the environment. Here it is the video access link:

https://www.youtube.com/watch?v=HwTKpR1R64E

Appendix A

Bill of Materials

#	Purchased item	QTY	Category	Unit Price ($)	Total Price USD
1	XYGStudy for Raspberry Pi SX1262 LoRa HAT 915MHz Frequency Band Spread Spectrum Modulation Supports Raspberry Pi 4 3 2 1 Series Boards Onboard CP2102 USB to UART Converter Interface for Arduino STM32	3	Communication	34.69	104.07
2	KAIWEETS Digital Multimeter TRMS 6000 Counts Ohmmeter Auto-Ranging Fast Accurately Measures Voltage Current Amp Resistance Diodes Continuity Duty-Cycle Capacitance Temperature for Automotive	1	Measurement Tool	35.99	35.99
3	SunFounder Ultimate Sensor Kit for Arduino R3 Mega2560 Mega328 Nano - Including 98 Page Instructions Book	1	Sensor Package	59.99	59.99
4	LED Power Supply Adapter 24V 10A - 240W AC/DC Power Adapter Transformer	2	Power Supply (for PAC/iPC)	45.99	91.98
5	Wireless Keyboard Mouse Combo, Cimetech Compact Full Size Wireless Keyboard and Mouse Set 2.4G Ultra-Thin Sleek Design for Windows, Computer, Desktop, PC, Notebook, Laptop - Silver	1	Keyboard-mouse (for PAC/iPC)	34.99	34.99
6	LoRa/GPS Long Range Transciever Shield 915 MHz (North America)	1	Communications	32	32
7	LoRa Long Range Transciever Shield 915 MHz (North America)	1	Communications	19	19
8	Botletics SIM7000 LTE CAT-M1 NB-IoT Cellular + GPS + Antenna Shield Kit for Arduino (SIM7000E)	1	Communications	65	65

#	Purchased item	QTY	Category	Unit Price ($)	Total Price USD
9	3-in-1 Global IoT SIM Card	1	Communications (LTE-M industry)	5	5
10	Readytosky 3DR Radio Telemetry Kit 915Mhz 100mW Air + Ground Module Open Source for Standard Version APM2.6 APM2.8 pixhawk 2.4.6 2.4.8 Flight Controller	1	Communications (Telemetry)	24.99	24.99
11	HiLetgo 3pcs ESP8266 NodeMCU CP2102 ESP-12E Internet WiFi Development Board Open Source Serial Wireless Module Works Great with Arduino IDE/Micropython (Large)	2	Short range Communications	12.99	25.98
12	seeed studio Raspberry Pi Camera Module V2 8 Megapixel 1080P, Standard CSI Compatible with Raspberry Pi 4, Raspberry Pi 3/3 B+, Raspberry Pi Zero/Zero W and Most Single Board Computer	2	Visual sensing	25.99	51.98
13	Obstacle Avoidance Sensor, Lidar Range Finder Sensor Module TFmini-s, 0.1-12m Lidar Detector Sensor Lidar Tiny Module Single-Point Micro Ranging Module with UART / I2C Communication Interface	1	Optical sensing	41.99	41.99
14	Bolsen 3pcs GP2Y0A21YK0F GP2Y0A21 10~80cm Infrared Proximity Distance Sensor	1	Sensing	25.99	25.99

#	Purchased item	QTY	Category	Unit Price ($)	Total Price USD
15	Smraza 5pcs Ultrasonic Module HC-SR04 Distance Sensor with 2pcs Mounting Bracket for Arduino MEGA R3 Mega2560 Duemilanove Nano Robot XBee ZigBee S03	1	Sensing	9.59	9.59
16	CQRobot VL53L1X Time-of-Flight (ToF) Long Distance Ranging Sensor for Raspberry Pi/Arduino/STM32. Accurate Ranging Up to 4m, I2C Interface. for Mobile Robot, UAV, Detection Mode, Camera, Smart Home.	1	Sensing	19.99	19.99
17	Gowoops GPS Module with TTL Ceramic Passive Antenna for Arduino Raspberry Pi 2 3 B+ MCU	1	Communications	20.99	20.99
18	Gowoops GPS Module with TTL Ceramic Passive Antenna for Arduino Raspberry Pi 2 3 B+ MCU	1	Battery for microcontroller board	18.99	18.99
19	LoStik by Ronoth - LoRa/LoRaWAN US 915mhz / Uses RN2903	1	Communications	46	46
20	2Sets 868MHz-915MHz SX1276 ESP32 WiFi Bluetooth LoRa Module Development Board with 0.96 OLED Display & Antenna Transceiver IOT for Arduino LoraWan Internet of Thing WIshioT	1	Communications	48.99	48.99
21	AMAX 5580 PAC/iPC (plus accessories)	1	Industrial Computer	1510.93	1510.93
	Grand Total Before Tax & Shipping				2294.43
	Grand Total After Tax & Shipping				2384.64