ASEE'S VIRTUAL CONFERENCE
At Home with Engineering Education
JUNE 22 - 26, 2020 #ASEEVC
Paper ID #28678

A Low Cost Kiosk for Student Learning of Human Machine Interface (HMI)

Dr. Larry Himes Jr, Purdue University-Main Campus, West Lafayette (College of Engineering)

Dr. Larry Himes, Jr.

Dr. Larry Himes, Jr. is a recent Ph.D. graduate from Purdue University. Working on a startup, Didactictron, Inc., manufacturing STEM education devices and kits for students. Has taught ECET undergraduate courses at Purdue University North Central in Westville, Indiana and EECT undergraduate courses at Ivy Tech Community College in South Bend, Indiana over the past six years. A couple of my STEM education kits, patented by Purdue University, were assembled and programmed by students at Ivy Tech. The goal is to make low cost and innovative products for STEM education classrooms. Dr. Larry Himes, Jr. grew up in Tyner, Indiana, a town with a population of about 200 in Northern Indiana. Started attending Purdue University in 1988 to complete the Associates degree and begin the Bachelor degree. Leaving in the middle of the Bachelor degree to start in industry. Then returning, after the stock market crash in the fall of 2008, to complete the Bachelor degree. Followed by the completion of the Master and Ph.D. degrees. Both the industrial experience and degrees pursued at Purdue University have included electrical and mechanical aspects. Dr. Larry Himes, Jr. is currently seeking a full time university faculty position.

A Low Cost Kiosk for Student Learning of Human Machine Interface (HMI)

Dr. Larry Himes, Jr. (KG9KV)

Abstract

The use of touch sensing devices is common in this day and age. Capacitive touch sensing is the most widely used, but there are resistive and reflective means as well. Low cost, simple circuitry, easy to implement and simple to program were the four factors considered for classroom use. A means of implementing the touch sensing in an application was another factor. The result was a kiosk to be assembled and programmed by Electrical Engineering Technology students. This kiosk is based on the P8X32A because it offers VGA video, playing WAV files, touch sensing, patron detection and control interfacing. The P8X32A is an octal core microcontroller that allows for eight separate programs to run simultaneously. There is no sharing of time on a single CPU as is the case with Real Time Operating System (RTOS). The touch sensing combined with the VGA and WAV player outputs makes for a more intuitive user experience. This paper outlines hardware and software used for this human machine interface (HMI) project.

Introduction

A kiosk is a small structure containing hardware providing means for user input as well as visual and audible feedback. The hardware converts the user input to control signals for manipulating a device, such as entering a security code and actuating an electric door lock. Early evidence indicates that rural kiosks can help villagers improve their economic standard of living by expanding livelihood options and empowering them with information, tools, goods, and services (such as education and healthcare).[1] In the ever-changing culture of today, it seems that the world continues to move toward "computer-facilitated self-service technologies" like ATMs, pay-at-the-pump gas stations, and self-checkout at grocery stores tend to unveil both supporters and critics of the idea.[2] A reporting kiosk is simply a stand-alone machine that resembles an ATM or it can be a dedicated computer where a probationer can report for required check-ins with the supervising probation agency.[3]

Instead of designing a hardware application and then choosing a microcontroller, it was decided to use a P8X32A because it had been used in a few past projects. The P8X32A is a 32-bit octal core microcontroller operating from a 5MHZ crystal multiplied to 80MHZ. That 80MHZ clock is divided by two with two clock cycles per instruction, or 20 million instructions per second (MIPS) is provided by each of the eight cogs. The 20 MIPS per cog for 8 cogs is 160 MIPS of processing power. The 8 cogs of the P8X32A allows it to run up to eight independent programs simultaneously. Cog0 starts up first and usually manages starting and stopping of the other cogs and copying global variables between the cogs. Another cog is used for VGA video rendering, one for the WAV player, one for monitoring the touchpads, one for transferring data on the CAN bus, one for monitoring the ultrasonic distance sensor and one for transferring data via the RF transmitter. Separating the functions of the kiosk among seven cogs, providing each function its own cog.

This helps to prevent interleaving of tasks in code when programming, such as Real Time Operating System (RTOS), that could lead to long delays. The programmer can write code for each task independently. Programming the P8X32A requires a USB to RS-232 converter device between the personal computer and P8X32A. A FT232BL provides the USB to RS-232 conversion and easily detected by most personal computers. After programming the kiosk, user interfacing functions are needed. A means of detecting if a user is present, greeting the user once present, providing audible and visual feedback to the user, sending data to handle the user request and detecting when the user has left are the basic functions needed on the kiosk. The kiosk functions are broken into the following categories:

- User inputs
- User feedback
- Data collection
- Execute user request
- Begin and end session

User inputs can be gesture, proximity and contact. Gesture when the user has only made hand movements, no physical contact with the kiosk. Proximity when a sensor is used to determine user distance from the kiosk. Contact when the user makes physical, non-destructive, contact with the kiosk.

User feedback can be audible, visual or physical. Audible when sound output is generated, such as music or sound effects. Visual with a video display or lighting effects or physical image movement is used. Physical when blasting the user with air, changing temperature around them, changing smell or adding vibration as means of feedback.

User data collection is accomplished using user input. Gesture, proximity and contact can be sensed, meaning data about the input produced. The collected data is then used to determine if a task is to be performed, user feedback is needed or user information needs to be stored for later reference.

User request execution could be unlocking a door, notifying someone of their presence, starting operation of a machine, stopping operation of a machine or resetting the kiosk.

User begin and end session would be detecting user presence and powering up the kiosk or detecting the user absence and shutting down the kiosk.

METHODOLOGY

Components: Hardware and Software:

- Parallax P8X32A 32-bit octal core microcontroller

- Google SimpleIDE C compiler
- FT232BL USB to RS-232
- USB A to USB mini B cable
- 8GB Micro Secure Digital card (micro SD Card)
- 15-pin D-sub VGA connector with resistor network for video
- Eight exposed / plated traces for touchpad user input
- Two 8 ohm speakers 1W each for audio output
- Audacity for conversion of sound files
- USB card reader for micro SD cards (PC to SD transfer)
- HC-04 Ultrasonic distance sensor module

First would be convert any sound files to the 16-bit PCM 32,000 samples per second monaural format needed for your application. Here two messages were used saying ”Welcome” and ”Goodbye”. https://www.text2speech.org/ is one website where you can enter text and have it render a WAV file for you. Then convert the WAV files to the format needed by the P8X32A using Audacity. After those files are converted to the P8X32A WAV format on your personal computer, those files need to be copied to the micro SD card. The file names need to be in the WAV player function followed by a delay equal to the time needed to play the entire WAV file. Place the micro SD card into the slot on the kiosk. The HC-04 ultrasonic distance sensor will be used for triggering the kiosk.

The Google SimpleIDE is used to program the P8X32A and is available under the GNU license at no cost from Google. The touch sensing operates at +3.3V DC with no potential of harm. The P8X32A has a built in VGA driver that can display about 25 columns by 24 rows in eight colors. Including the vgatext.h driver and using the dprint(vga, ”text here”); statement allows for characters to be displayed on the VGA screen. The video resolution is similar to the Commodore VIC20, an 80’s computer. Replacing standard text with inverted or highlighted text, can be used for visual feedback to the user indicating cursor location. Switching between the two rapidly could be used to indicate error or selection as well. The screen is cleared when no user is present and has a welcome screen when a user approaches. The P8X32A can play WAV files in 16-bit PCM at 32,000 samples per second monaural format.

Including the wavplayer.h driver and assigning the SD card control pins (P22 = MOSI, P23 = SCLK, P24 = MISO and P25 = CS) using the standard pins, allows for an 8GB micro SD card to be used for WAV file storage. Audacity is one of the better programs to use for converting other sound file formats to the 16-bit PCM 32,000 samples per second needed. The WAV files could be used for greeting a user, or instructing a user how to use the kiosk, playing music while the user is at the kiosk or a salutation as the user leaves the kiosk as examples. A HC-04 ultrasonic distance sensor is used for detecting presence of a user. It has a trigger input for sending the ultrasonic tone and an echo output providing a pulse width output in proportion to the distance of the person or object. The P8X32A offers touch detection that is arranged in a joystick configuration on this project. There are three touchpads: left, middle and right for function inputs. The joystick has up, right, down, left and center touchpads allowing the user to move the cursor around the screen.

Touch position is managed by the microcontroller which in turn inverts or highlights the option text, indicating where the cursor is on the screen as user feedback. A Local Interconnect Network (LIN) bus is implemented as one means of transferring data from the kiosk to a control circuit. The LIN bus speed is 19,200 baud 8N1 and first byte packet value of 85 for timing synchronization. The LIN bus could be used for controlling lighting in the kiosk, such as turning off when no user is present and turning on when a user walks in. Another is using a MICRF113 transmitter operating on the 433.050MHZ to 434.790MHZ Industrial, Scientific and Medical (ISM) frequency band. A SKY65162-70LF could be used after the MICRF113 to boost the kiosk transmitting range. The P8X32A provides a clock output in the range of 13.532820MHZ to 13.587180MHZ that will be multiplied by 32 times, and a data output control line for sending serial data.

A square-wave(18, 0, 13560000); command will output a 13.560MHZ clock on pin P18 of the P8X32A to the MICRF113 oscillator input pin. The 13.560MHZ is multiplied by 32 times at the MICRF113 to put out 433.920MHZ at +10dBm of RF power. A RF power output of +20dBm or less is needed to comply with FCC ISM band rules. The SKY65162-70LF provides about 20dB of gain, for an amplified RF power output of near +30dBm. RF levels above +20dBm will require FCC type acceptance or an Amateur Radio license. A copper trace antenna can be placed onto the PCB with the kiosk, if desired. Or, a PCB mounted SubMiniature version A (SMA) connector can be used for connecting an external antenna and coaxial cable. ISM band antennas are easily obtained from many vendors. The frequency range can cover 420MHZ to 450MHZ when operated by someone holding an Amateur Radio license.

The HC-04 ultrasonic distance sensor has a control pulse input called trigger, and a pulse width output called echo. The trigger input needs a logic high (+3.3V or +5.0V) pulse of at least 10 micro-second duration. This causes the HC-04 to send out 8 cycles of 40KHZ ultrasonic waves to bounce off of objects. The reflected 40KHZ tone causes the HC-04 to produce an output logic high for a duration proportional the distance of the object. The output pulse duration is in micro-seconds and needs to be divided by 74. This division converts the echo output pulse duration of the device to inches. Now that the functions of the kiosk have been introduced, let's move on to how those functions are used.

Monitoring the HC-04 echo output pulse is needed for good operation of the kiosk. If the echo pulse is longer than 550 micro-seconds for more than 20 seconds, then the kiosk needs to announce "Goodbye" and shutdown. When the echo pulse is shorter than 550 micro-seconds for more than 3 seconds, then the kiosk needs to announce "Welcome" and wakeup. Wakeup includes turning on the VGA display, lighting and possibly music or narration inside the kiosk area. While announcing "Welcome" a text message indicating that the user touch the control, be displayed. Then bring up a menu after the user has initially touched the controls.

One text selection on either side of the screen. The left side showing "Enter" and the right side showing "Exit". Highlight the text respective to the side the user has swiped toward. If the center touchpad is touched, then make action. "Enter" causes the solenoid latch to actuate while the kiosk announces "Welcome" and flashes "WELCOME" text on the screen. "Exit" causes the solenoid latch to de-actuate while the kiosk announces ”Goodbye” and flashes ”GOODBYE” on the screen. In either case, the kiosk shuts down 20 seconds after selection, and de-actuates the solenoid latch at shutdown, if it had been actuated.

Building

This project was built, programmed and implemented by students. Presence of a user needs to be detected first. This is done using the HC-04 ultrasonic distance sensor module. A trigger output from the P8X32A on P20 with high logic (+3.3V) level of 10 micro-seconds minimum is needed. The pulse-out(20, 10); function causes a 10 micro-second high pulse output on P20 to the HC-04. That high pulse causes the HC-04 to put out eight cycles of 40KHZ audio. The audio signal bounces off of an object and returning to the HC-04 sensor. Making the Echo output high after the trigger pulse is completed and then low when the reflected audio is received, creates a pulse width output. The Echo line is +5.0V logic whereas the P8X32A is +3.3V logic and that is the reason for the 133 ohm series resistor. A 1.7V difference between the two logic levels divided by 133 ohms for about 13mA maximum current limit. The P8X32A has a means of detecting a user between 2cm to 400cm, as specified in the datasheet for the HC-04 module.

This is about 3/4” to 13’ of detection range but only 8” or less will be used. The expression a = pulse-in(21, 1); is used for collecting the Echo pulse width in micro-seconds of time. That variable value is divided by 148 to indicate user distance in inches or left as a value of 1,200 or less to indicate a user is present. A detection delay, Echo value less than 1,200, of 8 seconds was chosen as a safeguard before power up of the kiosk. This prevents false startup to ignore those passing by. Then a 20 seconds delay for shutdown after the user leaves. This prevents false shutdown if the user is moving around in front of the kiosk while using it. The user distance monitoring is performed on a separate cog. Status of user presence is reported to the other eight cogs using a global variable. This allows the cog to repeatedly check for user presence without interfering with the kiosk functions.

User input is discussed next. P08, P09 and P10 are for the three touchpads above the joystick area. No input resistors are used as the 3.3V logic level does not pose any harm to the user. The touchpad is composed of three expose and plated traces. A center trace for touch sensing and an outer trace, on either side, providing ground reference. Three touchpads for buttons at the top and five touchpads in a joystick arrangement at the bottom provides eight points for user input. The eight touchpads each has its own I/O line of the P8X32A, P08 through P15. One cog can effectively manage only eight touchpads when using the pulse-out() and rc-time() functions in the compiler. Removal of the input resistor help speed up the touchpad responsiveness considerably, as was found during circuit testing. The pulse-out(15, 10); function indicates P15 putting out a high pulse (+3.3V) for 10 micro-seconds.

That is followed by reading the rc-time(15, 1); function value to a variable, such as a. The value of variable a indicates the time in micro-seconds for the trace to discharge. If this value is between about 250,000 to 12,000, then no touch was detected. If the value is below 12,000, then a touch was detected. It was also found that the pressure of the touch can be monitored, but was not implemented. A value about 12,000 indicates a light touch, whereas, a value of about 2,000 indicates pressure causing discomfort to the finger. A resistor with parallel capacitor could be placed between a touchpad and ground to lower the maximum rc-time() value, but no attempt has been made thus far. Suggested values would be 100K ohm and 100pF to begin. Faster touchpad monitoring would be possible if custom code is used but the results are good when using the compiler functions.

Video output is in Video Graphics Array (VGA) format from the P8X32A on port 0 or P0 through P7. P0 passes through a 255 ohm resistor to pin 14 of the Dsub-15 connector for the vertical sync output. P1 passes through a 255 ohm resistor to pin 13 of the Dsub-15 connector for the horizontal sync output. P1 passes through a 255 ohm resistor to the common end of a three resistor network. P2 passes through a 499 ohm resistor to the common end of the resistor network. And a 133 ohm resistor connected between the common end of the network to ground. The common connection is the analog Blue video signal output connected to pin 3 of the Dsub-15 connector. A separate three resistor network is needed for each of the video color outputs: Red, Green and Blue. Pin 9 of the Dsub-15 connector is supplied with +5.0V regulated at 500mA to power the monitor interface logic. Pins 5, 6, 7, 8 and 10 are tied to ground for completing the Dsub-15 connections. The vgatext.h include file, vga = vgatext-open(0); for video on P0 through P7 and dprint(vga,"Your Text Here"); are needed to implement the 8 pin VGA port on the P8X32A.

Audio outputs are managed from P26 and P27 outputs of the P8X32A. Each output has the same monaural output and both are based on 48KHZ Pulse Width Modulation (PWM). A 255 ohm resistor from one audio output to the junction of a RC low pass filter is needed as signal input. And a 0.1uF capacitor from the junction end of the resistor to ground is needed to complete the filter. The low pass filter junction point is connected to the positive side of a 10uF electrolytic capacitor, and the negative side to pin 3 (non-inverting input) of a LM386 audio amplifier chip. Pin 2 (inverting input) and pin 4 (ground) are tied to ground. Pin 6 is power to the LM386 that needs to be bypassed with a 220uF electrolytic capacitor. Another 220uF electrolytic capacitor with the positive lead connected to pin 5 (audio output) of the LM386, and negative lead connected to the positive side of an 8 ohm speaker. The negative side of the speaker connected to ground. This circuit is repeated for both audio outputs of the P8X32A.

Data output is managed in two ways. One way is over a bidirectional CAN bus style serial link and the other is a one way RF link using an Industrial, Scientific and Medical (ISM) band. A square-wave(18, 0, 13560000); command will output a 13.560MHZ clock on pin P18 of the P8X32A. This reference clock output is passed through a 255 ohm resistor to a 5.6uH fixed inductor of a LC low pass filter. The output side of the inductor has a 24pF RF capacitor to ground. This LC low pass filter output has a 24pF RF capacitor to the oscillator input of the MICRF113. The oscillator output of the MICRF113 has a 24pF RF capacitor to ground. The 13.560MHZ is multiplied by 32 times by the MICRF113 to put out 433.920MHZ at +10dBm of RF power. A 470nH inductor between the RF output of the MICRF113 and +3.3V supply is needed both to provide power to the transmitter circuit and to isolate RF noise from the power lead.

Another 24pF RF capacitor is needed from the RF output to the antenna or input of a SKY65162-70LF RF amplifier device. Only use the SKY65162-70LF if you or your group has authorization to use more than 100mW (+20dBm) of RF power. A RF power output of +20dBm or less is needed to comply with FCC ISM band rules. The SKY65162-70LF provides about 20dB of gain, for an amplified RF power output of near +30dBm. RF levels above +20dBm will require FCC type acceptance or an Amateur Radio license. A copper trace antenna can be placed onto the PCB with the kiosk, if desired. Or, a PCB mounted SubMiniature version A (SMA) connector can be used for connecting an external antenna and coaxial cable. ISM band antennas are easily obtained from many vendors. The frequency range can cover 433.050MHZ to 434.790MHZ for the ISM band and 420MHZ to 450MHZ when operated by someone holding an Amateur Radio license, or when FCC type accepted for a specific application.

CAN is a differential half-duplex serial communication interface. The MicroChip ATA6560 CAN transceiver requires +5.0V for operation and converts the RS-232 data from the P8X32A to CAN. A 255 ohm resistor between the RX output of the ATA6560 and P17 serial input of the P8X32A is needed. Using a baud rate of 115,200 with 8 data bits, no parity and 1 stop bit (8N1) is well within the 5 megabits per second limit of the ATA6560 transceiver. The CAN transceiver needs a 155 ohm resistor across the CANL and CANH outputs at the master and at the last slave node. A second ATA6560 CAN transceiver would be added to the end of a twisted pair of wires in another location for conversion of the CAN data back to RS-232. If only two nodes are implemented, the master and one slave, then a 155 ohm resistor is needed across the CANL and CANH lines at each end. A TPS929120 LED driver with 12-channel output and an ATA6560 CAN transceiver could be an additional slave node, without the 155 ohm resistor, to provide lighting control inside the kiosk area. Each output of the TPS929120 can handle 50mA maximum. One output could power a solenoid lock actuator relay or controlling a ventilation fan or all outputs for driving four RGB LEDs making attention grabbing lighting effects. An ATA6560 CAN transceiver with a microcontroller would allow up to 254 nodes and one master when not using CAN data protocol. But 127 nodes or less when using CAN interface software protocol of vendors. There is no power provided by the CAN interface, only data. Please keep that in mind when adding nodes.

External storage is needed because the P8X32A internal FLASH is filled with a SPIN interpreter and other services such as the SD card interfacing. P28 is the Inter-Integrated Circuit (I2C) bus bidirectional data line, referred to as SDA. P29 is the I2C bus clock line that is output only for timing on the I2C bus. A 24LC512 was chosen to allow for the standard 32KB of user program (firmware) storage and an additional 32KB of data storage. The data storage region will not be used by normal programming, but instead having I2C code to manipulate the upper 32KB of EEPROM storage. User care is needed to prevent overwriting the lower 32KB block and destroying the firmware. The micro SD card could be used for extra data storage as well. The variable designations: int DO = 22, CLK = 23, DI = 24, CS = 25; are needed for the wavplayer.h include file to access the micro SD card. And the sd-mount(DO, CLK, DI, CS); function is for mounting the micro SD card to access the FAT32 table and data files, such as WAV sound files.

Programming is accomplished using a FT-232BL USB to RS-232 adapter from Future Technologies. When using the Google SimpleIDE for the Propeller (P8X32A), the IDE provides both the C compiler programming environment and a terminal interface for monitoring operation of the device or reviewing data collected or feedback when users type information from the computer keyboard. The P8X32A I/O lines P28, P29, P30 and P31 are not provided to the user as to protect the most critical purpose of these lines, programming and startup loading. A FT-232R PCB module with header is available from China for about two dollars. An open collector generic NPN transistor, with emitter tied to ground, collector tied to the reset input of the P8X32A and a

RC circuit (high pass configuration) from the Data Terminal Ready (DTR) line from the PCB module. That RC circuit is a 0.1uF capacitor from the DTR line to the NPN transistor base and a 10K ohm resistor from the transistor base to ground.

Discussion

When the user was within 8” or 20cm of the kiosk it would turn on, announce ”WELCOME”, display ”Welcome” in white text on a blue background using the VGA monitor and turn on the lighting inside the kiosk. The next screen message requested that the user make a selection. The left selection was for opening the door lock to enter and the right selection was for latching the door lock when leaving. If the left touchpad of the joystick was touched, then the code for opening the door lock was sent and the kiosk announced ”Welcome” a second time. The default shutdown time of 20 seconds was used for both the user making a selection and the user absence from the kiosk. A digital code was sent down the CAN style bus to open the door lock for the 20 second waiting period. Another digital code was sent down the CAN style bus to close the door lock just before shutting down the kiosk.

Another important point need to be considered is that the height and space around the kiosk should be convenient for both standing and wheelchair users to access the keyboard and information.[4] Tasks of a kiosk were discussed including: detecting user presence, providing user feedback, acquiring user input data and providing action related to user input. The Google SimpleIDE C compiler is free for anyone in the world to download, that requires only a USB port for programming and terminal interface. These tools make it easier for students to program and receive feedback as they are developing a kiosk or HMI. The MICRF113 was not used in this project, but it was tested at +10dBm output with FM modulation using a WAV file on the micro SD card that was successful. The P8X32A works well as a kiosk engine and all functions worked as expected.

Kiosks can be used for other functions such as in law enforcement, banking, education, self-check-in at hotels (including programming the door key) and placing restaurant orders as examples. Kiosks are often located in probation offices, courthouses, or police departments, and they typically use biometric identification (e.g., handprint or fingerprint scan) to verify probationer identity.[3] It sounds like there will be even more kiosks to come. Will this technology become larger than necessary?

Conclusion

Understanding technology use and usability are critical skills for modern IT professionals.[5] This is true for all areas of Science, Technology, Engineering and Mathematics (STEM) to know what tools are available and how to use them, not only IT. Moreover, the participants were asked for their objective evaluation regarding the experience they had with the interface, whether they liked it or not and whether they found it easy to use or not.[6] Overall, students enjoyed this project in the past both the hardware assembly and building the code to add functions. Hopefully more will want to build a kiosk in the future. The components used in the first run of this project were through hole and for the second run were 0805 SMD or larger, SOIC and TQFP to allow manual soldering and PCB assembly by students. Student learning involves many facets but not limited to: building the physical hardware needed for the project, understanding how the hardware operates, writing code for manipulating the hardware, developing code to improve the user interface with the system and completing a project for an intended function.

Reference

[1] Kentaro Toyama, Karishma Kiri, Deepak Menon, Suneet Sethi, Joyojeet Pal, and Janaki Srinivasan. Pc kiosk trends in rural india. *E-Learning and Digital Media*, 3(3):317–324, 2006.

[2] Avery Moore, Kristyn Reinecke, and Megan Plekker. Kiosk welcome center. 2018.

[3] Eileen M Ahlin, Carol A Hagen, Michele A Harmon, and Scott Crosse. Kiosk reporting among probationers in the united states. *The Prison Journal*, 96(5):688–708, 2016.

[4] Mahmut Ekşioğlu. User experience design of a prototype kiosk: a case for the istanbul public transportation system. *International Journal of Human–Computer Interaction*, 32(10): 802–813, 2016.

[5] Corey Brady, David Weintrop, Ken Gracey, Gabby Anton, and Uri Wilensky. The ccl-parallax programmable badge: Learning with low-cost, communicative wearable computers. In *Proceedings of the 16th Annual Conference on Information Technology Education*, pages 139–144, 2015.

[6] Mahmut Ekşioğlu, Handan Güler, Fulya Terzi, Haydar Sinan Yıldırım, and Buse Yücel. Uxd for a prototype campus information kiosk. 2018.