Microcontroller Systems Project 2DX3

Theme Report - 3 (Act)

Instructor: Dr. Athar, Dr. Doyle, and Dr. Haddara

Gurleen Kaur Rahi - L01 - rahig - 400377038

April 9, 2023

Theme

Act

A key component of creating outstanding intelligent systems that can interact with their environment in the field of artificial intelligence is the **Act** theme. An intelligent system is made to observe its environment, process information, and take appropriate action in response to that information. What distinguishes a good intelligent system from other computer programs and enables it to carry out activities that would otherwise be impossible is its capacity to interact with the environment. This is especially helpful in the workplace and for specialized applications, when intelligent systems are better equipped than humans to handle difficult jobs with accuracy, efficiency, and safety. Researchers and engineers may build machines that have the potential to revolutionize a wide range of industries, from manufacturing and logistics to healthcare and robotics, by emphasizing the subject of **Act** in the development of superior intelligent systems.

For example, in Lab 7, the microprocessor could manage numerous activities at once and respond in real-time to outside events thanks to interrupts. This shows how an intelligent system may communicate with the environment by taking input from sensors and providing outputs to operate objects. In order to measure distances, a Time-of-Flight sensor was employed. The distance data collected was then used to start interrupts and generate outputs through GPIO pins.

In Lab 8, to determine an object's position in space and operate a stepper motor in a particular direction, readings from the TOF sensor were employed. This was an example of how an intelligent system could use information from sensors to decide how to act in the real world. The microprocessor could interface with the PC and the sensor using the I2C and UART communication protocols, which allowed the system to efficiently collect and process data.

Numerous industries, including manufacturing, healthcare, transportation, and agriculture, use intelligent systems like those shown in Lab 7 and Lab 8, which have many applications in these fields. Intelligent systems in manufacturing can be used to monitor production lines, automate procedures, and enhance quality control. For instance, intelligent systems can employ sensors to find product flaws and initiate corrective operations to raise quality and lower waste. In the field of healthcare, intelligent systems can be used to check drug compliance, keep tabs on patient health, and spot early disease symptoms. For instance, wearable technology can gather and analyze information on a patient's vital signs, enabling medical practitioners to spot possible health problems before they become life-threatening.

Intelligent systems have the potential to increase efficiency and safety in the transportation sector. For instance, autonomous vehicles can utilize sensors and algorithms to recognize road conditions and react accordingly, lowering the chance of accidents and enhancing traffic flow. Intelligent systems in agriculture can be used to track crop development, increase fertilization and irrigation, and raise agricultural output. Sensors, for instance, may gather information on soil moisture, temperature, and nutrient levels, enabling farmers to modify their methods and maximize crop development. Intelligent systems are generally very adaptable and beneficial in a

variety of industries and applications because of their capacity for data collection and processing, decision-making, and interaction with the physical world.

Background

Dealing with this theme calls for the use of the TOF sensor, stepper motor, microcontroller, and onboard LEDs. Manufacturing, automation, and robots are just a few industries that often use these components. The basis for the lab centers on the idea of interrupt-driven programming and how significant it is in practical applications. Microcontrollers and embedded systems frequently have to handle a number of simultaneous activities, including reading data from sensors, controlling actuators, and interacting with other devices.

In these situations, conventional polling-based programming techniques are frequently ineffective since they necessitate ongoing input verification and may result in delays in responding to external events. Instead of spending resources on pointless polling, interrupt-driven programming enables the microcontroller to respond instantly to external events and manage numerous activities at once. In this lab, the goal of interrupt-driven programming is to create a square wave with a 25% duty cycle by employing periodic interruptions in place of the polling approach. This shows the effectiveness and responsiveness of interrupt-driven programming as well as how it may be used for real-time hardware control.

Additionally, the lab investigates the usage of I2C and UART as communication protocols to connect with a TOF sensor and operate a stepper motor using the sensor's data. This demonstrates how interrupt-driven programming can be used to do complicated tasks in real time by taking inputs from sensors and providing outputs to control devices. The lab serves as a practical example of interrupt-driven programming's use in real-world applications to manage numerous processes concurrently, react instantly to outside events, and control hardware in real time.

Theme Exemplar

Lab 8 Milestone 2

By combining the TOF sensor with earlier stepper motor code to obtain various distance measurements from a single 360^{o} horizontal scan, Milestone 2 uses an intelligent approach to interact with the environment. The system can see the spatial relationships between items in its environment and modify its behavior by turning these measurements into x, and y coordinates and plotting them using a program like Excel. The approach employed in this milestone is clear, thorough, and complete since it uses the TOF sensor in conjunction with previously written stepper motor code to record eight or more distance measurements spaced 45 degrees apart. To visualize the data, the measurements are then transformed into x, and y coordinates and plotted using software like Microsoft Excel. By gathering information about the environment and using it to guide its activities, this technique enables the system to actively interact with its surroundings.

The system can accurately capture and analyze distance data in real time, as shown by the validation findings of this method, and it may modify its behavior as a result. One of the key components of an intelligent system's capacity to interact with its surroundings is its capacity to detect and react to changes in the environment.

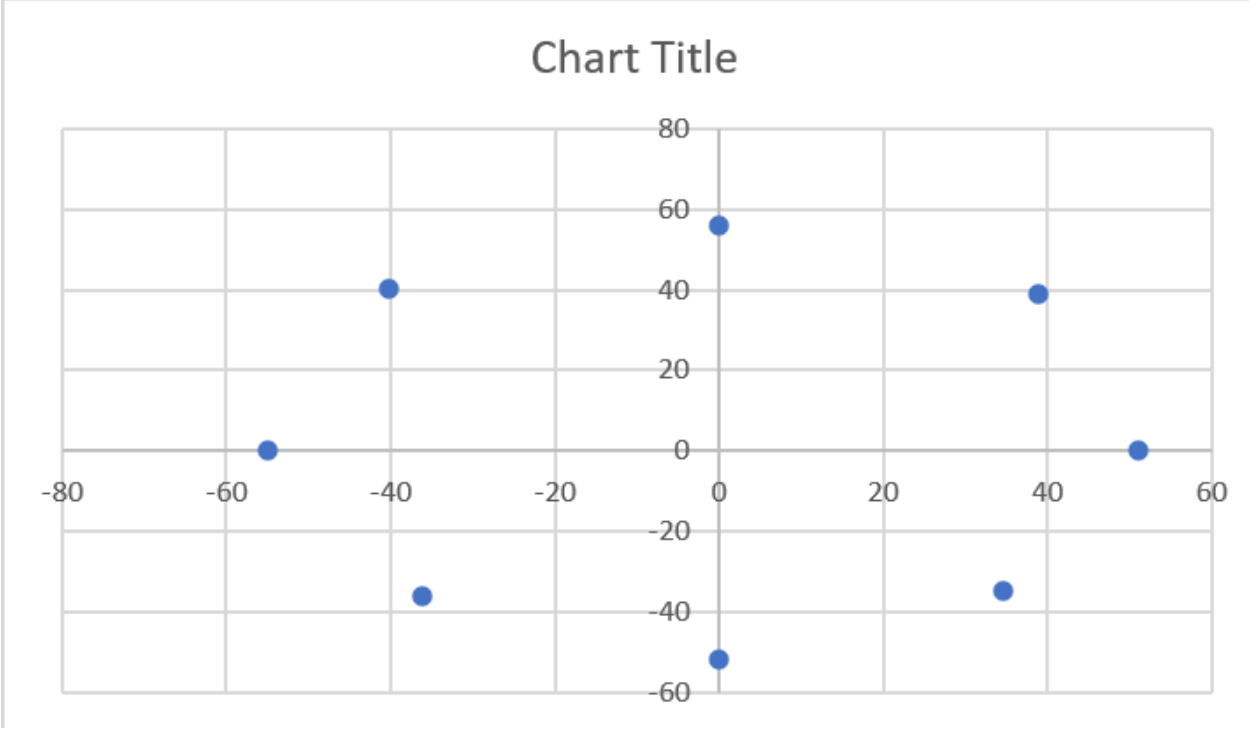

Figure 1: Plot generated by Excel

In general, the lab work examples offered in Milestone 2 are closely related to the concept of interacting with the environment, and the method employed and validation results gained are described in a clear, complete, and detailed way.

Lab 7 Milestone 1

The next achievement demonstrates the topic of the act by using the microcontroller to output a square wave and flash an internal LED. Polling, which was traditionally employed in labs, is faster than this method. This approach is more effective since the microcontroller may do other activities while an interrupt is in progress.

```
  // Part II: Arm and configure at Timer module side
  TIMER3_CTL_R = 0x00000000;              // (step 1) Disable timer3 during setup (Timer stops counting)
  TIMER3_CFG_R = 0x00000000;              // (step 2) Configure for 32-bit timer mode
  TIMER3_TAMR_R = 0x00000002;             // (step 3) Configure for periodic mode
  TIMER3_TAPR_R = 0;                      // (step 4) Set prescale value to 0; i.e. Timer3 works with Maximum Freq = bus clock freq (120MHz)
  TIMER3_TAILR_R = (period*120)-1;        // (step 5) Reload value (we multiply the period by 120 to match the units of 1 us)
  TIMER3_ICR_R = 0x00000001;              // (step 6) Acknowledge the timeout interrupt (clear timeout flag of Timer3 )
  TIMER3_IMR_R = 0x00000001;              // (step 7) Arm timeout interrupt

  // Part III: Enable interrupt at Processor side
  NVIC_EN1_R = 0b1000;                   //Enable IRQ 35 in NVIC
  NVIC_PRI8_R = 0x40000000;              //Priority 2
  EnableInt();                           // Global Interrupt Enable

  // Part IV: Enable the timer to start counting
  TIMER3_CTL_R = 0x00000001;              // Enable timer3
}


//This is the Interrupt Service Routine.  This must be included and match the
//interrupt naming convention in startup_msp432e401y_uvision.s (Note - not the
//same as Valvano textbook).
void TIMER3A_IRQHandler(void){
  TIMER3_ICR_R = 0x00000001;               // acknowledge timer3 timeout
  FlashLED1(25);                           // execute user task -- we simply flash LED
```

Figure 2: Code from Milestone 1

By using interruptions to generate a square wave rather than the more conventional polling technique, this milestone illustrates the idea of an intelligent system interacting with the environment. The resulting program may generate a square wave with a period of 1 second and a duty cycle of 25% by implementing a periodic interrupt that triggers every 1 second and developing an interrupt service routine (ISR) to flash an onboard LED for 250ms and output to a GPIO pin. As shown by confirming the timing of the square wave by connecting the GPIO pin to a scope probe and analyzing the associated pulse and interrupt periods, this enables the intelligent system to actively generate a signal and interact with its environment.

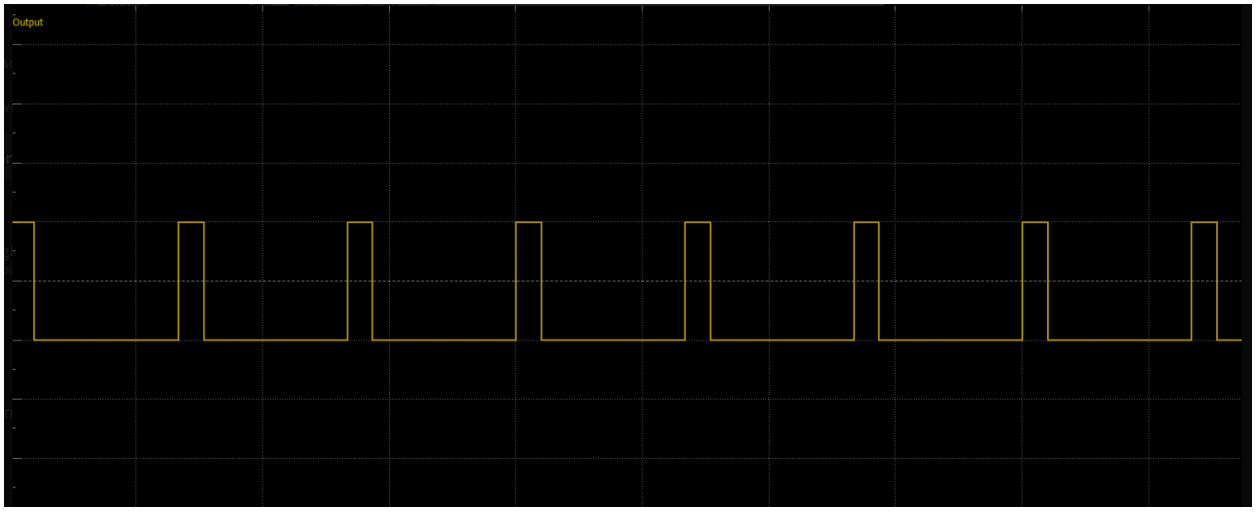

Figure 3: Square-wave Waveform

In this milestone, a square wave output illustrates the idea of an intelligent system engaging with the environment by actively generating a signal and reacting with it. The square wave output is a periodic signal with a set period and duty cycle that alternates between high and low states. The system may demonstrate its capacity to interact with and react to outside events in a timely and accurate manner by creating this square wave using interrupts and tracking its timing using a

scope probe. Intelligent systems that are intended to interact with and adapt to their surroundings must possess the capacity to actively generate a signal and respond to outside events. This is a crucial component of the ‘Act’ concept in intelligent systems.

Debugging Exemplar

Lab 7 Milestone 1

The "stepwise refinement" methodology is a simple, thorough, and applicable debugging technique for Milestone 1. In order to identify any potential problems, this method entails breaking the problem down into smaller steps and evaluating each step as it is implemented. Let's look at an instance where the square wave isn't formed properly to illustrate this technique. By adding a debug print statement to the ISR that shows the current time, we can first use the stepwise refinement method to confirm that the interrupt is occurring once every second. The next step, which involves adding another diagnostic statement to the ISR to confirm that the LED is flashing for 250ms, can be performed if this step is successful. We can use further debugging techniques, such as adding more print statements or using a debugger to step through the code and isolate the problem, if any problems are encountered at any point in the process. Once the problem has been located, we can update the code and run another test using the stepwise refinement method to make sure all of the components are operating properly.

Synthesis

Lab 8 Milestone 1

It is crucial to remember that establishing serial UART communication is a technique for an intelligent system to communicate with the external world by exchanging information in order to show that you have a correct knowledge of how this milestone relates to the theme. In this instance, the VL53L1X sensor is using the UART protocol to transmit the model ID and module type to the PC. The system can read the required data from the sensor and send it to the PC by utilizing the specified code and particular API function calls. It's crucial to adhere to the wiring diagram provided for the lab to guarantee that the components are connected correctly. Due to the internal pull-up resistor on the VL53L1X PCB, AD2 and the pull-up resistors must be removed. This achievement demonstrates how a smart system may exchange information with its environment using communication protocols.

Reflection

Milestones 1 and 2 in the lab illustrate how intelligent systems and microprocessors can interact with the environment. The use of interrupts to produce a square wave and interact with the environment is demonstrated in Milestone 1. The program demonstrates the system's capacity to interact intelligently with its surroundings by being able to actively generate a signal and respond to outside events in a prompt and effective manner. Intelligent systems created to interact and adapt to their environment must possess the capacity to actively emit a signal and respond to

outside events. The system's capacity to actively interact with its surroundings by acquiring information about the environment and using it to guide its actions is demonstrated in Milestone 2. The system may adjust its behavior by capturing and analyzing distance data in real-time using the TOF sensor and stepper motor code. An essential component of an intelligent system's capacity to interact with its surroundings is its capacity to detect and respond to environmental changes.

Both milestones' approaches are thorough, succinct, and informative, giving readers a thorough knowledge of how intelligent systems and microprocessors can be used to interact with and adapt to their surroundings. The system's correctness and efficiency in recording and analyzing data in real time are also shown by the validation findings. Overall, by demonstrating how intelligent systems and microprocessors can actively generate signals, respond to external events, and adapt their behavior based on data obtained from the environment, the lab examples in Milestones 1 and 2 help students understand the acting theme in these systems.

Reference

[1] *Theme Report - McMaster University*. [Online]. Available: https://avenue.cllmcmaster.ca/d2l/le/content/512368/viewContent/4110708/View. [Accessed: 10-Apr-2023].