# 2DX3 – Theme Report 3: Act

By: Vraj Patel

I hereby declare that the work presented in this report is original, and any external sources or materials used have been duly cited in accordance with the accepted standards of academic integrity and ethical research practices.

# Theme

The central theme of the forthcoming report will revolve around the foundational notion of "Act" within smart systems. The "Act" represents the culminating action undertaken in the process of smart system operation. Initially, the system engages in "Observation" by gathering relevant data, followed by "Reasoning" to determine the appropriate course of action based on the collected data. Finally, the system executes a specific action derived from the reasoning process. An exemplary instance of "Act" within smart systems is showcased by the Automatic Emergency Braking system implemented in vehicles. Drawing insights from collected data and informed reasoning, this system swiftly responds to critical situations, such as imminent collisions with vehicles ahead or sudden braking by preceding vehicles. In such scenarios, the Automatic Emergency Braking system promptly executes braking maneuvers, surpassing human reaction times, to mitigate potential accidents effectively.

# Background

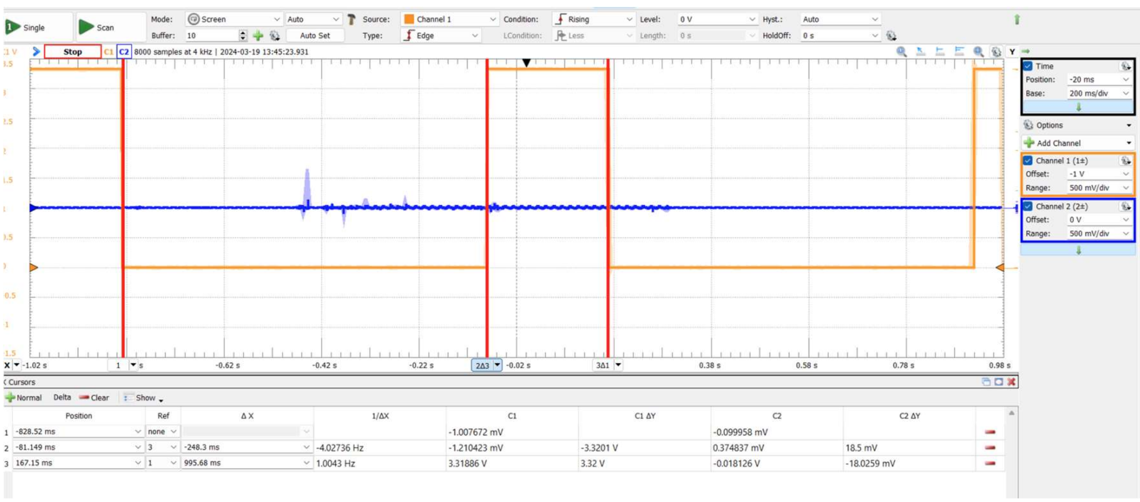
The "Act" phase within the smart system operational cycle represents the pivotal stage where the system executes actions based on the insights gathered and processed during the "Observe" and "Reason" phases. Arguably the most crucial of the three stages, "Act" empowers the system to translate data into tangible actions. Without this critical phase, the system would merely accumulate valuable data without the ability to act upon it. Analogously, envision conversing with a virtual assistant like Siri, which, devoid of the "Act" component, merely registers input without effecting any subsequent action. Even within the context of the project, if the motor doesn't spin, then the special mapping of the hallway is impossible. This is exactly why the "Act" phase is an important attribute of a smart system. Furthermore, the effectiveness of a smart system is often gauged by its performance in the "Act" phase, influencing consumer decisions regarding system adoption or purchase.

Compared to the other phases of Reason and Observe, the implementation of the Act phase offers a higher degree of flexibility due to its diverse range of applications. In our project, communication between the three devices enables the collection of data and instructs the motor to initiate spinning, although this approach may vary. In the realm of IoT (Internet of Things), data is typically stored, and actions are executed based on the accumulated data within an IoT network [1]. Additionally, a conventional method of prompting a system to "Act" is through remote control. For instance, when watching TV, users can switch channels using a remote control, and the system responds accordingly based on the input received. This illustrates the varied approaches available for implementing the Act phase in different systems and contexts.

# Theme Exemplars

**Theme Exemplar #1:**

The following exemplar is from Lab 7: Interrupts and Event-based Programming, specifically Milestone 9.1, where it was required to generate a square wave using interrupts instead of polling. The square wave was to have a period of 1s with a duty cycle of 25%, and would also have to flash an onboard LED for 250ms.



*AD2 Graph Milestone 9.1*

```
void PortJ_Interrupt_Init(void)
{

    FallingEdges = 0; // Initialize counter

    GPIO_PORTJ_IS_R = 0;      // (Step 1) PJ1 is Edge-sensitive
    GPIO_PORTJ_IBE_R = 0;     //          PJ1 is not triggered by both edges
    GPIO_PORTJ_IEV_R = 0;     //          PJ1 is falling edge event
    GPIO_PORTJ_ICR_R = 0x02;  //          Clear interrupt flag by setting proper bit in ICR register
    GPIO_PORTJ_IM_R = 0x02;   //          Arm interrupt on PJ1 by setting proper bit in IM register

    NVIC_EN1_R = 0x00080000; // (Step 2) Enable interrupt 51 in NVIC (which is in Register EN1)

    NVIC_PRI12_R = 0xA0000000; // (Step 4) Set interrupt priority to 5

    EnableInt(); // (Step 3) Enable Global Interrupt. lets go!
}

void GPIOJ_IRQHandler(void)
{
    I2C_Send1(0x29, 0x00);
    FlashLED(1);
    GPIO_PORTJ_ICR_R = 0x02; // Acknowledge flag by setting proper bit in ICR register
}
```

*Code for Milestone (IRQ handler & Initializations)*

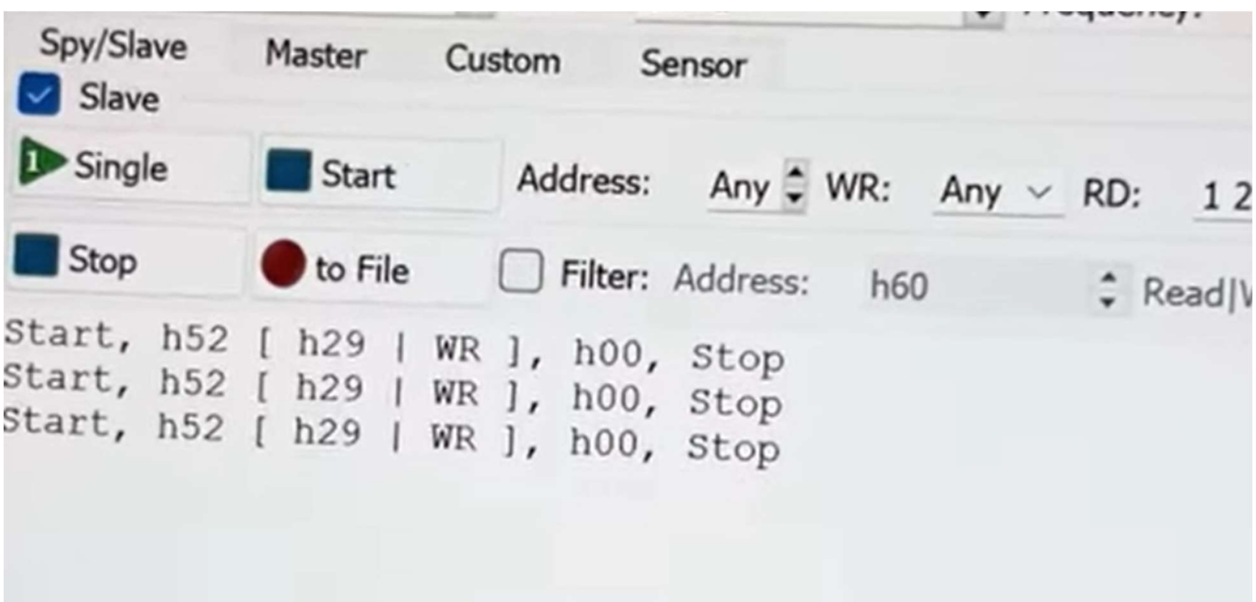
## Debugging Exemplar

In the debugging process outlined in the theme report, my partner and I encountered an issue with the data transmitted during the execution of Lab 8: Collecting Distance Data, Milestone 10.1. Upon identifying the discrepancy, we initiated the debugging phase to rectify the problem. Our approach involved closely examining the content of the I2C packet to understand why our Time-of-Flight (ToF) sensor was not functioning as expected. To facilitate this, we utilized the provided code to inspect the I2C contents thoroughly. This meticulous analysis allowed us to pinpoint the root cause of the issue and implement an effective solution.

```
void GPIOJ_IRQHandler(void){
  GPIO_PORTJ_ICR_R = 0x02; // Clear
  I2C_Send1(0x29,0x0);
```

*Code for I2C package*

Subsequently, the Analog Discovery 2 (AD2) was connected, and upon successful transmission of the address, it wrote a hexadecimal 0, indicating a successful transmit operation. However, in cases where a garbage value was received, it became apparent that there was an issue with the data transmission. Following the detection of the garbage value, a thorough examination of the wiring revealed that the sensor had been improperly connected. Upon correcting the wiring, a successful data transmission was achieved, as depicted in the subsequent image.




*Corrected Version of Milestone after Debugging*

## Synthesis

In Lab 7, the task involved generating a square wave using interrupts, which was accomplished and validated through the waveform displayed on the AD2. The square wave exhibited a 250ms high period, as intended. This achievement directly correlates with the thematic concept of "Act" within intelligent systems. In the realm of intelligent systems, the "Act" phase denotes the execution of specific actions in response to received data or triggered events. Similarly, in this milestone, the implementation of a periodic interrupt serves as the trigger event, prompting the system to generate a pulsing square wave. This process demonstrates the system's ability to actively respond to predefined conditions or events by executing programmed actions autonomously. By incorporating interrupts to drive the square wave generation, the system showcases its capacity to dynamically perform tasks based on established criteria, thereby embodying the essence of the "Act" theme in intelligent system design.

## Reflection

The completion of Lab 8: Collecting Distance Data, Milestone 10.1, and Lab 7: Interrupts and Event-based Programming, particularly Milestone 9.1, not only enhanced my understanding of the "Act" phase within intelligent systems but also provided valuable insights into the broader concept of intelligent systems as a whole.

In Lab 8, the debugging process allowed me to gain practical experience in troubleshooting issues related to data transmission, which is a fundamental aspect of intelligent systems. By closely examining the content of the I2C packet and analyzing the provided code, I developed a deeper understanding of how data flows within a system and how devices communicate with each other. This hands-on experience highlighted the intricate nature of intelligent systems and underscored the importance of effective communication protocols in ensuring seamless operation.

Similarly, Lab 7 provided me with insights into the active response mechanisms employed by intelligent systems. By implementing interrupts to drive the generation of a pulsing square wave, I gained practical experience in leveraging system capabilities to execute predefined actions autonomously. This experience emphasized the dynamic nature of intelligent systems and the importance of real-time responsiveness in various applications.

Overall, both lab experiences contributed significantly to my understanding of intelligent systems by providing practical insights into key concepts such as data transmission, communication protocols, and real-time responsiveness. By engaging in hands-on debugging and implementation tasks, I developed a deeper appreciation for the complexities of intelligent systems and gained valuable skills that are essential for success in this field.

# References

[1] "Intelligent systems: What are they, how do they work and why are they so important," Algotive, https://www.algotive.ai/blog/intelligent-systems-what-are-they-how-do-they-work-and-why-are-they-so-important (accessed Apr. 7, 2024).