Independent Study on Self-Programming of Droplets by Over the Air Programming and Analysis of CSMA based Droplet-Droplet Communication

Bhallaji Venkatesan Kulathu Sundaram

May 14, 2017

An independent study of the droplets- a swarm robotics platform, was performed under Prof. Nikolaus Correll, from a three-way perspective of firmware, electronics and droplet-to-droplet communication. The self-programming of the droplets was implemented based on TDMA-based IR communication transfers. The design of CSMA-based IR communication was analyzed, aiming to optimize the time for data transfer between droplets. This independent study was performed at the Correll Lab of the University of Colorado, Boulder as part of the Master's coursework in Electrical Engineering with Professional Masters in Embedded Systems.

1 Introduction

The Droplets are an experimental platform for large-scale swarming research. The robots are capable of sensing IR and RGB intensities, actuation using vibration motors, and IR communication. Further features include- unlimited running time without stopping to recharge (they get power from the floor through their legs), omni-directional motion (6 linear directions and turn-in-place), directional communication with user-settable communication range from 1cm to 1m, range and bearing sensing, true-color, full-body illumination, self-righting, droplet-to-droplet reprogramming with an expandable RAM / Flash. The hardware is comprised of actual robots, and a test bed / floor on which experiments are run. The embedded software layer is what we use to write and compile on-board code for the droplets. Each robot is equipped with an Atmel xmega-128a3u microprocessor capable of running Embedded-C. The independent study was geared towards improving the droplet-to-droplet reprogramming and analyzing a prospective CSMA-based Droplet-to-Droplet communication. The droplets communicate with each other through the IR communication protocol operating at 38kHz carrier frequency.

Since Droplets, as a platform has the potential to transition into a commercial product owing to increasing interests in commercial low-cost swarm platforms, the factors such as ease of use, affordability and faster communication are of utmost necessity from a scalable viewpoint. This independent study focuses on improving the ease of use and an analysis of an approach towards increasing the speed of the droplet-to-droplet communication. This study involved self-programming a swarm of droplets from a master droplet, which could potentially reduce the programming durations of multiple robots. This creates a use case where the user can program multiple robots at the same time, enhancing the viability of droplets as a standalone platform. The implementation aspects, varying design constraints and challenges faced during the implementations are discussed in this report. Further, the droplets use TDMA-based IR communication with 29 TDMA time slots, divided between 63 droplets in the current implementation. For a scalable future, such a design has the possibility to create a lot of 'dead air', where none of the droplets would use the communication channels effectively, leading to exorbitant delays. Thus a possibility of CSMA/CA based approach is discussed and a prospective implementation of the same is proposed. The entire experiment is performed in the Correll Lab at The University of Colorado Boulder under the guidance of Prof. Nikolaus Correll and John Kingler. This study was performed alongside Mr. Rahul Yamasani, Master's Student in Electrical Engineering at The University of Colorado Boulder.

2 Approach

This independent study involves 4 phases: 1. Understanding the concepts behind droplets 2. Implementing basic droplet functionalities 3. Implementing the self-programming functionality 4. A brief study on CSMA / CA as a possible implementation of a droplet-to-droplet communication

2.1 Droplet Concepts

Dealing with the droplet platform involves hardware and software aspects. The hardware platform involves usage of the Atmel-ICE J41800048908 debugger and droplet programmer built on ATxmega128A3U. The software and the firmware for the system developed on ATMEL Studio 7.0 platform involved a brief study of the scheduler, communication and other interfaces of the software architecture .

2.2 Basic Droplet Functionality Implementation

Firstly in the process of learning how the droplet functions, sequential color patterns through the RGB LEDs were implemented with the `set_rgb()` condition scheduled in the loop function that gets called periodically by the scheduler.

As the next step, simple directional movements of droplets were performed using `move_steps()`, `walk()`,`stop_move()` and `is_moving()` functions.Finally, coordinated movements between 2 droplets were implemented using the above functions and IR communication functions including `get_rnb()`, `check_collisions()`, `broadcast_rnb_data()` to communicate between two droplets for precise, coordinated movements.

Implementing these basic test functionalities provides the knowledge about the necessary firmware and software architecture along with their judicious usage for serial debugging techniques during the droplets' runtime.

2.3 Self-Programming Implementation

2.3.1 Concepts

To understand the concept of self-programming, the fundamental concepts of programming a hardware platform and how programs boot from flash memory to start execution of the functionality needs to be studied.

The concepts of boot-loader loading the application program from the flash memory was well understood. In the context of droplets, self-programming can be viewed as a slave droplet receiving the 'hex' file over the wireless IR protocol, to be flashed into its flash memory. The hex file that is received, will be stored in the SRAM of the ATxmega128A3U by default. The process of transferring the hex file contents from the SRAM to the Flash memory and enabling the boot-loader to execute the new flash contents as the application can be precisely defined as self-programming of the droplets. To understand the entire process, I am presenting the necessary technical description of self-programming and flash memory organization of ATxmega128A3U as show in Fig. 1 and Fig. 2.

The SPM commands (Self-Programming) are used for writing and reading the application code from the flash memory. The important aspect to be taken into consideration while working with the SPM commands is, they should be executed from bootloader with ".boot" attribute for them to function, since it is the role of the Bootloader to Read-Modify-Write the application section of the flash memory. The SPM commands used for the erasing a particular application page , erasing and writing into a flash application page, separately writing into an application page, loading a flash page into a buffer are `SP_EraseApplicationPage()`, `SP_EraseWriteApplicationPage()`, `SP_WriteApplicationPage()`, `SP_LoadFlashPage()`.

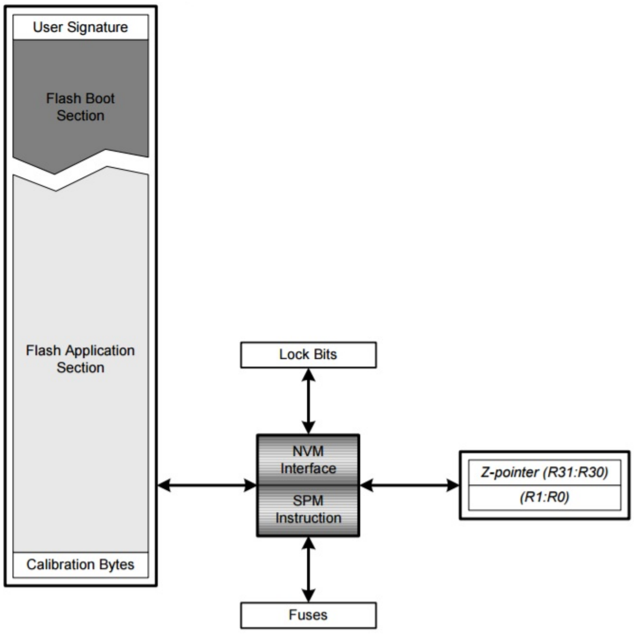

Figure 1: Overview of Self-Programming in ATxmega128A3U.

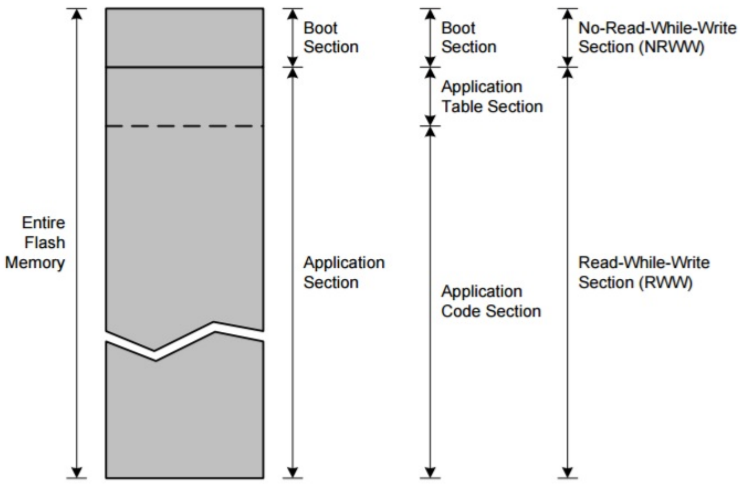

Figure 2: Flash Memory Organization in ATxmega128A3U.

2.3.2 Implementation

The Self-Programming functionality in the droplets was implemented with an incremental approach. Firstly, the `SP_LoadFlashPage()` function was used to determine if the flash contents are read properly. The correctness of the result was verified by the comparing with original 'hex' file contents, that were used for flashing the code. Secondly, the `SP_EraseWriteApplicationPage()` function was used to write a different data to the wrapper section (the application memory is divided into sections of Wrapper, User code and Application section set in the linker settings) of the hex file. This operation failed, because the first page of flash write was always erased as shown in Figure 3.The reason for this was later found to be an errata of the self-programming driver implementation in ATxmega128A3U where, "The EEPROM and Flash cannot be written while reading EEPROM or Flash, or while executing code in Active mode". The work-around for this as suggested by the Atmel forums indicated to "Enter IDLE sleep mode within 2.5uS (Five 2 MHz clock cycles and 80 32 MHz clock cycles) after starting an EEPROM or flash write operation. Wake-up source must either be EEPROM ready or NVM ready interrupt." After performing this work-around, the functionality worked as expected and the verification was performed as done in the first approach as mentioned above.

The Next step involved two droplets, a Master droplet, transmitting hex code to a slave droplet programmed with RED LED glow. The Master droplet transmitted a hex code equivalent to GREEN LED glow to the slave droplet using the over the air IR Communication and self-programming was performed.

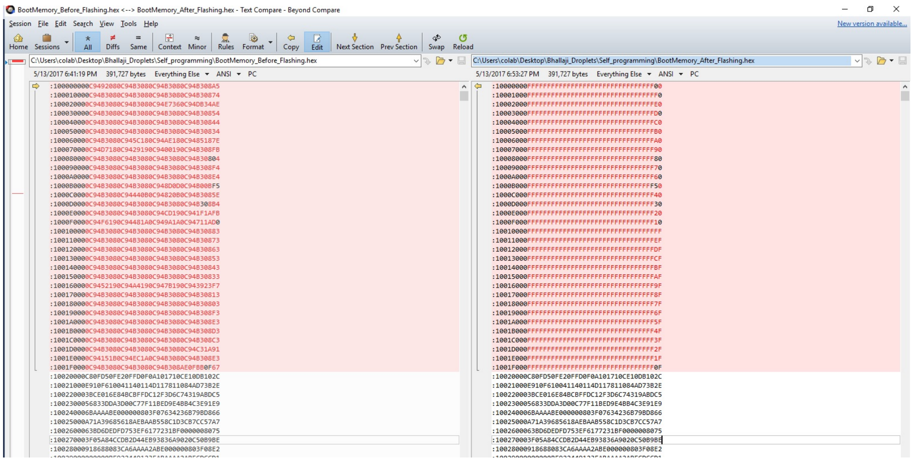

Figure 3: First Page Erased Upon Flash write at Wrapper Function.

2.4 A Study on CSMA/CA

In a multi-robot system like Droplets, an efficient carrier transmission protocol is of utmost need, for reliable and fast communication. Different communication schemes for such a system include, Aloha, Slotted Aloha, CSMA and TDMA protocols. Since TDMA is already being used for the Droplets platform, the need for the CSMA arises from the fact that, the earlier implementation is not efficiently using the carrier frequency and has the potential to cause exorbitant delays in a scalable architecture. In the current scenario, 29 time slots is divided between 63 droplets, which can potentially have "dead air" where none of the droplets are using some slots and the other droplets can still not use the channel to transmit. Thus a need for another efficient protocol arises. Upon comparison of CSMA/CA, Aloha and Slotted Aloha along with the Figure 4 , shows that CSMA/CA has a better Access Probability and Collision Immunity when compared to Aloha and Slotted Aloha.

In CSMA/CA, as soon as a node receives a packet that is to be sent, it checks to be sure the channel is clear (no other node is transmitting at the time). If the channel is clear, then the packet is sent. If the channel is not clear, the node waits for a randomly chosen period of time, and then checks again to see if the channel is clear. This period of time is called the backoff factor, and is counted down by a backoff counter. If the channel is clear when the backoff counter reaches zero, the node transmits the packet. If the channel is not clear when the backoff counter reaches zero, the backoff factor is set again, and the process is repeated.

The proposed implementation of CSMA/CA in droplets involves the concepts of RTS/CTS to avoid long collision. In this case, prior to the data transmission the sending node will send a RTS packet to announce the upcoming transmission When the destination node receives the RTS it will send a CTS packet after a particular interval. The sending node is allowed to transmit its data packet only if it receives the CTS packet correctly. The suggested implementation has to be performed in the `perform_ir_upkeep()` function, which handled the core of IR transmissions.

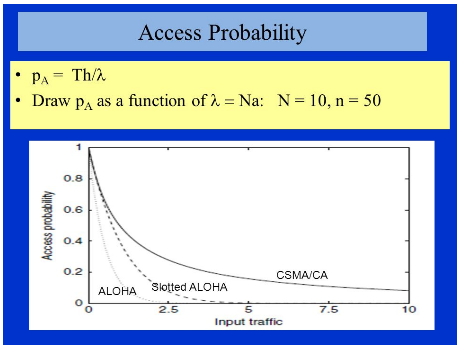

Figure 4: Comparison of Aloha vs Slotted Aloha vs CSMA/CA. (Source : Internet)

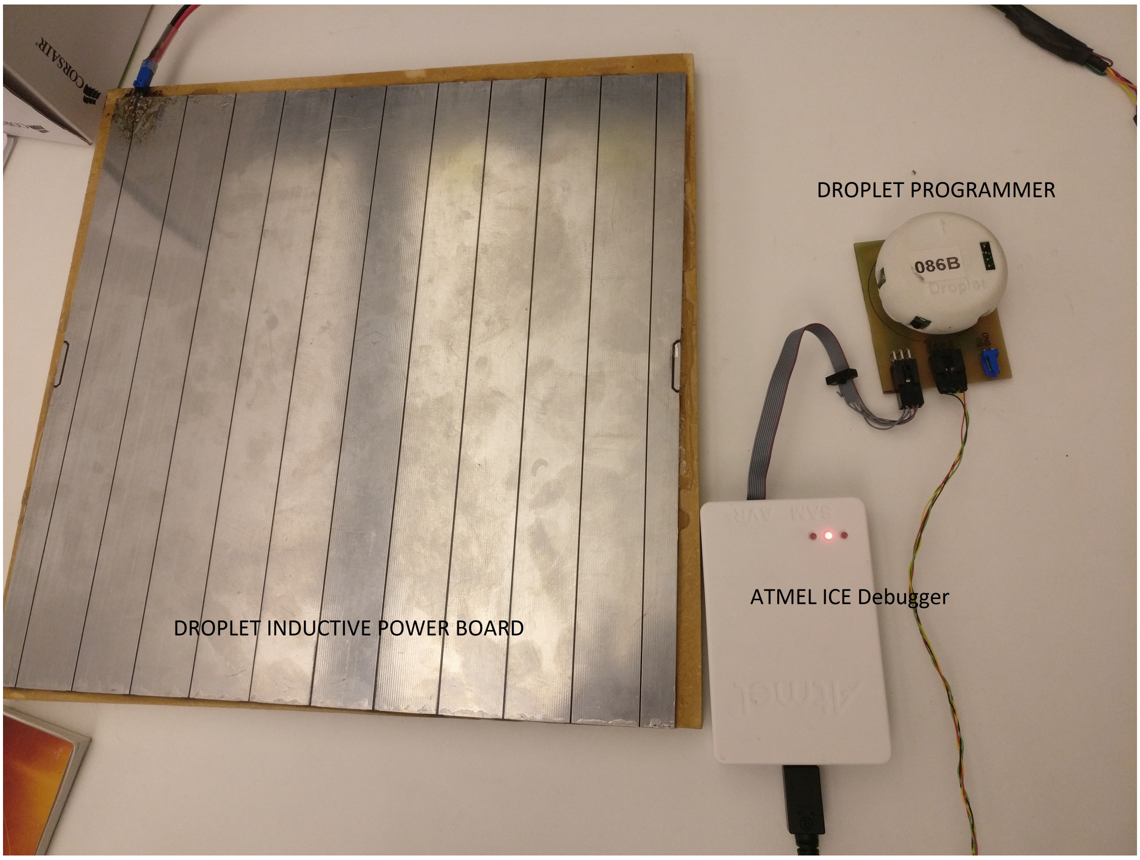

Figure 5: Before vs After flashing through self-programming.

3 Experiments

The experiments involved an incremental approach towards the self-programming of the droplets. The experimental setup as shown in Figure 5, includes a Test bed, which provides the power source to the droplets, the droplet robots, droplet programmer and the Atmel ICE debugger.

The final demonstrated test involves a Master and a Slave droplet. The master droplet is programmed with a Blue LED glow program , that transmits the hex code of a Green LED glow program through TDMA based IR transmission. The slave droplet is programmed with with a Red LED glow program, that polls for reception of IR messages to perform self-programming of the received hex file and execute the code.

The entire experimental result with the slave droplet changing its LED glow from Red to Green upon IR reception is shown in the Figure 6 along with the hex file contents.

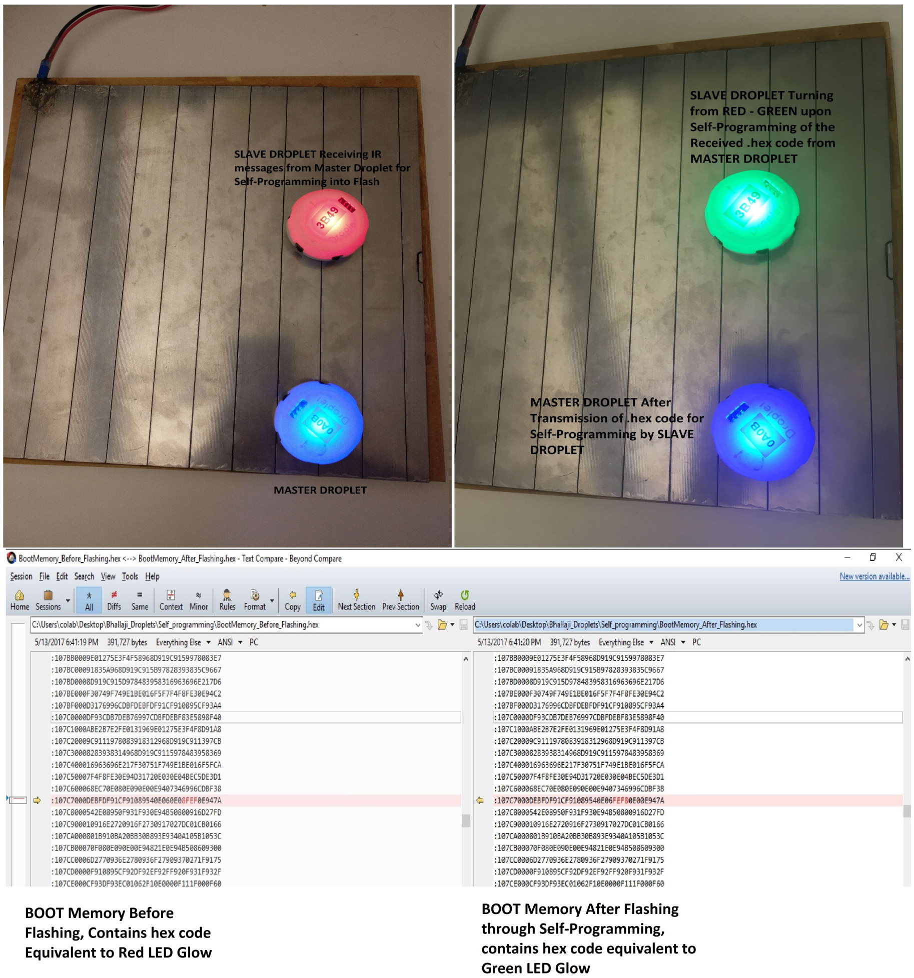

Figure 6: Before vs After flashing through self-programming.

4 Discussion and Future work

The Self-Programming of the droplets can be improved with error-detection and error-control of various use-cases of flash page writes and different sections of the Application section of the Flash memory. Further, the development of request and re-transmission of missing part of code transfer for a master-slave combination can be performed for efficient and faster retrieval rates. The CSMA/CS implementation is definitely an option to be considered for the future work due to potential efficiency in the optimal channel usage amongst the droplets.

5 Conclusion

The analysis and improved functionality of self-programming of the droplets, as a swarm robotics platform was performed. Further, a demonstration of the self-programming through Over-the-Air IR communication was depicted with associated results. The study on the Droplets enables us to realize that, the platform has a great potential to become a commercial product, with enhanced functionalities of Self-Programming for the ease-of-use and CSMA/CA based IR Communication for improved reliable and faster communications for the droplet-droplet interactions.