**BIRDS-5 OBC**

**Software Manual**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Revision Log** | | | | |
| **Sl. No.** | **Date** | **Person responsible** | **Changes** | **Remarks** |
| 1 | 2022-07-07 | Keenan |  | Initial Release |

**Table of Contents**

[1.0 Background 3](#_Toc108442557)

[2.0 Command and Data-Handling 3](#_Toc108442558)

[3.0 Command Format 7](#_Toc108442559)

[4.0 BIRDS-5 Software Initialization Procedure 8](#_Toc108442560)

[5.0 Software Flow Diagrams 17](#_Toc108442561)

[MAIN PIC Flow Diagrams 17](#_Toc108442562)

[FAB PIC Flow Diagrams 19](#_Toc108442563)

[RESET PIC Flow Diagram 20](#_Toc108442564)

[COM PIC Flow Diagram 21](#_Toc108442565)

[Mission Boss PIC Flow Diagram 22](#_Toc108442566)

# 1.0 Background

BIRDS-5 project consists of two (2) identical 1U CubeSats and one (1) 2U CubeSat. Each satellite performs the same missions (except PINO in the Japanese satellite) and operates at the same Uplink and Downlink frequencies. The satellites will remain off until the release from ISS and there will be no RF transmission until 30 mins after release from ISS. The satellites will turn ON after release from ISS and the antennas will deploy after 30 mins. Each satellite will be programmed to transmit beacon at different times to avoid overlapping of signals. And each satellite will be assigned a unique call sign. The CubeSats has a total of 10 MCUs as follows:

Table 1: BIRDS-5 MCUs

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No** | **MCU Type** | **MCU Specs** | **MCU Name** |
| 1 | PIC | PIC16F1789 | RESET PIC |
| 2 | PIC | PIC 18F67J94 | MAIN PIC |
| 3 | PIC | PIC16F1789 | COM PIC |
| 4 | PIC | PIC16F1789 | FAB PIC |
| 5 | PIC | PIC16F1789 | MB PIC |
| 6 | STM32 | STM32F446RE | ADCS MCU |
| 7 | PIC | PIC18F67J94-I/PT | SF-WARD MCU |
| 8 | RPI | RPI 0 | IMG-CLS MCU |
| 9 | RPI | RPI 0 | MULT-SPEC MCU |
| 10 | Xilinx | - | PINO |

The three satellites in the BIRDS-5 Project are:

1. ZimSat-1
2. PearlAfricaSat-1
3. Taka

# 2.0 Command and Data-Handling

The OBC has 5 main purposes:

1. Send the Continuous Wave (CW) Beacon
2. Collect, store, and transmit Housekeeping and Mission Data
3. Analyze the Uplink commands received from Ground Station
4. Execute mission commands and store mission data
5. Monitor the general status of the satellite and survive space environment.

The BIRDS BUS uses three dedicated microcontrollers (Reset PIC, COM PIC, Main PIC) for each of the three major subsystems, i.e., the EPS subsystem, the Communication subsystem, and the command and data-handling subsystem. An additional microcontroller (FAB PIC) collects the electrical power information as a monitoring device. And finally, the Mission Boss PIC controls command and data flow between the missions and the BIRDS BUS.

The BIRDS BUS uses a simple 8-bit microcontroller family for several reasons. BIRDS projects are educational projects, so a simple and easy-to-use microcontroller is better than a high-performance microcontroller. The performance of a BIRDS BUS microcontroller may be lower compared to the microcontrollers used in other CubeSat projects, especially those with more practical missions. These simple microcontrollers are, however, sufficient for an educational project with a simple mission and no attitude control.

**Diagram

Description automatically generated**

Figure : BIRDS-5 OBC Block Diagram

The Block Diagram in the figure above shows a schematic diagram of the data-handling system. For easy training with a microcontroller, the BIRDS BUS uses just one family of microcontroller. The PIC microcontroller series was selected because of its flight heritage with many CubeSat projects. It would be better to use the same PIC16F1789 microcontroller throughout all the subsystems; however, the Main PIC requires high speeds and enough computational power compared to the other microcontrollers, because it handles all the satellite data. There can be a data transmission bottleneck between the BUS system and the Mission system, and there must be adequate programming memory to handle various mission system data in future projects. Because of that, a more powerful PIC microcontroller, PIC18F67J94, is used for the Main PIC. It has enough programing memory at 128 [kBytes], and it is easy to modify its programs following the requirements of the mission

system. Also, it supports a maximum 64-[MHz] clock speed for data handling. The Main PIC has a 10-pin digital interface with the mission system. These digital interface pins can be configured to serial interfaces.

Non-volatile memory is required for the data storage of the data-handling system. The BIRDS BUS has four non-volatile memories. Two of them are dedicated storage for the Main PIC and the COM PIC, and the other two are shared memory with a multiplexer. One of the shared memories is between the Main PIC and COM PIC, and the other is between the Main PIC and the mission system. Each of the shared memory multiplexers is controlled by the Main PIC. The BIRDS BUS uses a simple UART serial interface for the regular interfaces between microcontrollers. It is easy to use but its speed is limited to 115,200bps. Shared memory supports large amounts of data transfer when the speed of the serial interface is insufficient. The BIRDS BUS uses only one type of flash memory as its non-volatile memory, a SPI interface NOR-type flash memory of 1 [Gbit] capacity. That is sufficient for a CubeSat if it has ordinary mission objectives. Because only one type of flash memory is used, a common library code for memory handling is available for the coding work. Not just for the non-volatile memory, but also for the serial interface, the BIRDS BUS uses just two common serial interface protocols. There are many kinds of serial interfaces for an embedded system, but only UART and SPI serial interfaces are used for the BIRDS BUS.



Figure : Ring Network

Because the BIRDS BUS is a distributed system for data handling, it needs well defined interfaces between the three major microcontrollers. Those three microcontrollers are connected to each other by UART for the primary interface, and construct a very simple pseudo ring network, as shown in the figure above.

Regular messages are transmitted between the microcontrollers in this simple ring network. The Reset PIC controls the electrical power supply for the entire CubeSat system. If other microcontrollers fail to send acknowledgements to the regular messages, the Reset PIC can reset the microcontroller power to force a power reset. These power controls should be very reliable, and the Reset PIC has very simple and clear programming code to minimize trouble. Because of the low probability of reset through the Reset PIC, it also keeps the satellite time data. Each of the satellite’s electronics parts, however, has the possibility of failure in orbit because of the single-event effect due to radiation. The Reset PIC is not an exception. In the case of Reset PIC failure, a simple external watchdog is attached to the Reset PIC to recover from the failure. The three major microcontrollers need to act as a combined data-handling system, and that requires time synchronization in many cases. Each microcontroller has its own primary clock source with a dedicated oscillator. However, one 32.768-[kHz] oscillator in the OBC is used as a common clock source for all three microcontrollers. The common clock source becomes the secondary clock source of each microcontroller to create a timer interrupt at the same time between the three microcontrollers. This simultaneous timer interrupt simply synchronizes the timing of the data-handling activities, and the satellite time management becomes much easier because of this common clock source. This synchronized data handling is especially useful for the regular messaging work of the UART ring network. In the BIRDS BUS, only the COM PIC handles communication with the ground station. It exchanges data with the Main PIC through the UART ring network or the shared flash memory. The UART interface is sufficient for small amounts of data; however, larger data such as image data needs to use the shared flash memory for efficiency.

Usually, two kinds of data are transferred to the COM PIC from the Main PIC. One is housekeeping data, the basic information about the satellite’s condition, and the other is the mission data from the mission system. Data is transmitted to the ground station by UHF transceiver. The COM PIC takes commands from the ground station and sends most of the commands to the Main PIC for further processes. Also, the COM PIC has its own flash memory to keep the data. A simple command to download the data does not need to be processed by the Main PIC, so it can be directly processed by the COM PIC with a rapid response. The Main PIC handles all the satellite data. It collects power supply system information from the FAB PIC and Reset PIC. The Main PIC is the data bridge between the bus system and the mission system. All data from the mission system comes to the Main PIC first through the serial interface of UART, SPI, or shared flash memory. As mentioned, most of the commands from the ground station are handled by the Main PIC too, and the Main PIC also controls one kill switch (the FAB PIC independently controls another kill switch to minimize risk).

# 3.0 Command Format

The Ground Station Command Format for BIRDS-5 is shown in the table below. It is 14-bytes long, from Header to CRC. It should be noted that BIRDS-5 command format is slightly different from previous BIRDS:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Header** | **Sat ID** | **Command Format** | **Command ID** | **Reservation Time** | **Mission Command Data** | | | | | | | **CRC** | | |
| **00** | **01** | **02** | **03** | **04** | **05** | **06** | **07** | **08** | **09** | **0A** | **0B** | **0C** | **0D** |

* **Header (1 byte)**- Leading byte for the command signal. Set by the GS Software.
* **Satellite ID (1 byte)**- Designated for each BIRDS-5 satellite. It can be chosen as any unique value for each satellite.
* **Command Format (1 byte)** - Determines which PIC the command is designated (0xA0 - MAIN PIC, any - COMM PIC, 0x33 - RESET PIC, etc.).
* **Command ID (1 byte)** - Determines which command to execute. The upper 4 bits describe which mission MCU the command is for the lower 4 bits describe what command to execute.
* **Reservation Time (1 byte)** - Designates the reservation time for specific mission. Delays command execution by the specified time.
* **Command Data (7 bytes)** - Describes the command data to be included in the command.
* **CRC (2 bytes)** - Cyclic Redundancy Check for error checking

An example camera capture command to the Multispectral Camera 1 on ZIMSAT-1: 11A0290000000000000000

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Header** | **Sat ID** | **Command Format** | **Command ID** | **Reservation Time** | **Mission Command Data** | | | | | | | **CRC** | |
| **-** | **0x11** | **0xA0** | **0x29** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **0x00** | **-** | **-** |

The Header and CRC are auto generated by the GS Software. SAT ID in this case refers to ZIMSAT-1. Command Format for MAIN PIC. Command ID shows a MCAM1 command. No reservation time so it will execute immediately. Command data is 0x00 as it is unneeded for the capture command.

An example Store and Forward Real-time Uplink Command to TAKA: 33A05100AB050106100105

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Header** | **Sat ID** | **Command Format** | **Command ID** | **Reservation Time** | **Mission Command Data** | | | | | | | **CRC** | |
| **-** | **0x33** | **0xA0** | **0x51** | **0x00** | **0xAB** | **0x05** | **0x01** | **0x06** | **0x10** | **0x01** | **0x05** | **-** | **-** |

The SAT ID now specifies a different satellite. The Command Format remains the same (to MAIN PIC). The Command ID now specifies an S&F command. The Mission Data can be anything within the 7 bytes

# 4.0 BIRDS-5 Software Initialization Procedure

**Software Upload Sequence**

The final software for the above mentioned MCUs shall be loaded into the CubeSats before delivery to JAXA in the following sequence and shall be verified by following checklist.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Action | Procedure /Check Point | Sat1 | Sat2 | Sat3 | Remarks |
| 1 | With the satellite OFF, load RESET PIC software with COM PIC [RESET\_PIC\_FM - COM ON immediately] turning ON immediately using Pickit4 and Mission boss pic software | The beacon should start as soon as Reset PIC is programmed. Check Mission PIC uart monitor software version |  |  |  |  |
| 2 | Remove RBF and connect pickit3 to Load COM PIC FM software | Check the following in the software:   1. Software version 2. Satellite ID   Use GS radio and CWGET software to decode the beacon. Note following in decoded beacon:   1. Call sign 2. Satellite ID |  |  |  |  |
| 3 | Load MAIN PIC Software | Connect to serial monitor and note the following:   1. Software version 2. Check the addresses, flags and sensor data from FAB, Reset and ADCS MCU |  |  |  |  |
| 4 | Put RBF and load Reset PIC turning on after 30 mins [RESET\_PIC\_FM] | The beacon shouldn’t start after the reset PIC is programmed |  |  |  |  |
| 5 | Load FAB PIC FM using Pickit4 |  |  |  |  |  |
| 6 | Load Reset PIC again using Pickit4 |  |  |  |  |  |
| 5 | Remove RBF and turn ON the satellite. |  |  |  |  |  |
| 6 | Stop the automatic mission and change the Antenna Deployment flag to 4 | 1. Send command “0xAB” for stopping the high sampling mission 2. Send command “0x6A” for resetting all the flags to 0 3. Send command “0xB4” for changing the antenna deployment flag 4. Send command “0x88” for checking the flags. All the flags must be 0 except antenna deployment flag which should be 4. |  |  |  |  |
| 7 | Turn **OFF** the satellite and be ready for the final functionality confirmation tests |  |  |  |  |  |
| 8 | Turn **ON** the satellite | 1. Check the serial monitor for addresses, flags and sensor data from FAB, Reset and ADCS PIC 2. Automatic Mission should start running (HSSC) |  |  |  |  |
| 9 | Wait 30 minutes, the COM PIC should turn **ON and antenna should deploy** | Check the CW GET software for beacon after 30 minutes |  |  |  |  |
| 10 | Wait for 1.5 hours more to let the Automatic mission execution to complete\* | After a total of 2 hours, the automatic mission should stop and downlink 10 packets every after CW |  |  |  |  |
| 11 | Perform CAM Mission | Use GS software and perform the following FT:   1. Send a command to perform HSSC/HK download 2. Send a command to perform CAM mission 3. Send a command to download CAM mission data and check the downloaded data packet |  |  |  |  |
| 12 | Perform all MBP Missions | Use GS software and perform the following FT:   1. Send a command to perform APRS    1. using handheld radio, transmit APRS message. ACK and beacon should be received |  |  |  |  |
|  |  | 1. Send a command to perform SFWARD    1. using handheld radio, transmit message |  |  |  |  |
|  |  | 1. Send a command to download SFWARD data and check the downloaded data packet (transfer to MBP, transfer to MAIN PIC) |  |  |  |  |
| 13 | Perform ADCS Mission | Use GS software and perform the following FT:   1. Send a command to perform ADCS mission 2. Send a command to download ADCS mission data and check the downloaded data packet |  |  |  |  |
| 14 | Check the operation of kill switch and un-kill switch | Use GS software and perform the following FT:   1. Send 5 commands to operate the kill switch. Listen for the ‘click’ sound in the satellite, check the serial monitor for CW type1, and HK data. 2. Send a command to undo the kill switch. Listen for the ‘click’ sound in the satellite, check the serial monitor for CW type1, and HK data. 3. Send a command to reset the satellite. Check the serial monitor. |  |  |  |  |
| 15 | Erase mission boss memory | send the following commands to the satellites using GS: 0X 00 A0 D1 00 A0 00 00 00 00 00  0X 00 A0 D2 00 A0 00 00 00 00 00  0X 00 A0 D3 00 A0 00 00 00 00 00  0X 00 A0 D4 00 A0 00 00 00 00 00  0X 00 A0 D5 00 A0 00 00 00 00 00  0X 00 A0 D6 00 A0 00 00 00 00 00  0X 00 A0 D1 00 A0 00 00 00 00 00  0X 00 A0 DF 00 00 00 00 00 00 00  0X 00 A0 DF 00 02 AA FD 57 00 00  0X 00 A0 DF 00 04 00 FC 00 00 00  0X 00 A0 DF 00 05 55 FA AB 00 00  0X 00 A0 D4 00 A5 00 00 00 00 00  0X 00 A0 DF 00 07 FF 00 00 00 00 |  |  |  |  |
| 15 | Erase MAIN PIC memory by pressing “EE” from the serial monitor | Check in the serial monitor for the following address:   1. Flag address: 00040000 2. Reservation Table : 00050000 3. Satellite Log : 00060000 4. CAM : 00080000 5. HK : 00620000 6. CW : 044A0000 7. ADCS : 04720000 8. MBP : 06660000 9. HSSC : 06670000 10. Address write counter : 00000001   Check in the serial monitor for the following flags:   1. Passes days : 0 2. Reserve check : 0 3. Kill Flag Main : 0 4. Kill Flag FAB : 0 5. Auto HSSC : 0 6. Auto CAM : 0 7. Auto MBP : 0 8. Auto ADCS : 0 9. Antenna Deploy Attempt : 0 10. Uplink success flag : 0 |  |  |  |  |
| 16 | Turn OFF satellite |  |  |  |  |  |
| 17 | Turn **ON** satellite | The flag of HSSC, CW address, HSSC address, HK address, Satellite log address should be changed |  |  |  |  |
| 18 | Erase MAIN PIC memory by pressing “EE” from the serial monitor | Turn off the satellite as soon as the memory is erased. |  |  |  |  |
| 19 | Turn OFF satellite |  |  |  |  |  |
| 20 | Charge the satellite for at least 4 hours |  |  |  |  |  |

Table. 2

**Upload procedure**

Following points shall be noted while uploading the software for the FMs:

1. The software loader shall be accompanied by another person to verify the software version
2. The second individual shall take note of the software version loaded and key features
3. Same individuals shall load the program for all three (3) FMs
4. The Satellite shall be turned off (step 19) as soon as step 18 is completed in Table 2.
5. The satellite shall not be turned ON ever after sequence 19.

With complete verification conducted using the table 2 the satellite shall remain off until it is released from ISS.

**Abbreviated procedure for software initialization**

This procedure can be followed if Main PIC, COM PIC and FAB PIC were programmed and tested using the final version of each software.

Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Action | Procedure /Check Point | Maya2 | Guarani | Tsuru | Remarks |
| 1 | Load Reset PIC software | Reprogram through FAB using PICKit |  |  |  |  |
| 2 | Turn **ON** the satellite | Check the serial monitor for addresses, flags and sensor data from FAB, Reset and ADCS PIC |  |  |  |  |
| 3 | Erase mission boss memory | send the following commands to the satellites using GS: 0X 00 A0 D1 00 A0 00 00 00 00 00  0X 00 A0 D2 00 A0 00 00 00 00 00  0X 00 A0 D3 00 A0 00 00 00 00 00  0X 00 A0 D4 00 A0 00 00 00 00 00  0X 00 A0 D5 00 A0 00 00 00 00 00  0X 00 A0 D6 00 A0 00 00 00 00 00  0X 00 A0 D1 00 A0 00 00 00 00 00  0X 00 A0 DF 00 00 00 00 00 00 00  0X 00 A0 DF 00 02 AA FD 57 00 00  0X 00 A0 DF 00 04 00 FC 00 00 00  0X 00 A0 DF 00 05 55 FA AB 00 00  0X 00 A0 D4 00 A5 00 00 00 00 00  0X 00 A0 DF 00 07 FF 00 00 00 00 |  |  |  |  |
| 4 | Erase MAIN PIC memory | Send the command “EE” from the serial monitor. Turn OFF immediately after “Memory erase done” appears and countdown start |  |  |  |  |
| 16 | Verify memory has been properly erased | -Turn ON satellite until FLAGS status, memory address and HK data is printed in the serial monitor then Turn OFF the satellite  -Check in the serial monitor for the following address:   1. Flag address: 00040000 2. Reservation Table : 00050000 3. Satellite Log : 00060000 4. CAM : 00080000 5. HK : 00620000 6. CW : 044A0000 7. ADCS : 04720000 8. MBP : 06660000 9. HSSC : 06670000 10. Address write counter : 00000001   Check in the serial monitor for the following flags:   1. Passes days : 0 2. Reserve check : 0 3. Kill Flag Main : 0 4. Kill Flag FAB : 0 5. Auto HSSC : 0 6. Auto CAM : 0 7. Auto MBP : 0 8. Auto ADCS : 0 9. Auto SFWARD : 0 10. Antenna Deploy Attempt : 0 11. Uplink success flag : 0 |  |  |  |  |
| 20 | Charge the satellite for at least 4 hours |  |  |  |  |  |

# 5.0 Software Flow Diagrams

## MAIN PIC Flow Diagrams

Diagram

Description automatically generated

MAIN Program

Chart

Description automatically generated Diagram

Description automatically generated

Execute Command

HK Collection

The program flow for MAIN PIC begins in the main program loop where we setup the registers, variables and interrupts that will be used in the main program. Once this setup function is completed, we enter the main loop. In this loop, the first step is checking if the antenna deployment flag is less than 4, if it is then we will attempt deploying the antenna and increment the flag. After this check, we check if the RESET FLAG is raised by RESET PIC. If reset flag is high, this indicates that RESET PIC is going to reset the satellite. MAIN PIC will prepare for reset by saving all flags, addresses and registers. After this reset check, we will check if a command was received from the PC or from the COMM PIC. If a command was received, we then check weather it was a command to be reserved by checking the reservation time. If there is no reservation time, then the command is executed immediately. In the” Execute Command” flow diagram, we see a series of checks to determine which MCU the command is destined to. The command is forwarded to the designated MCU and then executed based on the command ID. The next check in the main loop is to determine if a reserved command is present and weather or not it is time to execute the command. If the reservation time has passed, the RSV FLAG will be raised, and the reserved command will be executed. Next, we check weather a mission is on, and if the mission has been on for longer than 15 minutes. All mission switches will be turned off if any mission was on for longer than 15 minutes. Finally, we conduct the housekeeping data collection if 90 seconds have elapsed since we last collected data. In the HK collection, we first reset the counter, then loop the address locations in the flash memory for HK and CW to verify that the data is not overlapping with other mission data. We collect FAB, RESET and ADCS data and then combine them to form the HK and CW from this data. We check if the COMM PIC needs the CW data and forwards it if needed. If a mission is active, we will save the HK and CW to Shared COM Flash and MAIN Flash. If it is not active, we can save to SCF, MF and Shared Mission Flash. Once HK collection is completed, we return to the top and begin the main loop again from the antenna deployment check.

## FAB PIC Flow Diagrams

Diagram

Description automatically generated

FAB Main loop is relatively simple as it collects data on behalf of MAIN PIC. FAB begins by setting up all data and variables. The first check is to determine whether the KILL FLAG is greater than 4. If it is then we engage the kills witches for OBC and itself, essentially crippling the satellite. This is needed in the final stages of the satellite lifetime. The next check is weather the MAIN PIC is requesting data, if it is, then we will collect the voltage, current and temperature data of the battery and solar panels and forward it to MAIN PIC. The final two check are to do with incrementing or resetting the KILL FLAG. After these checks the loop is started again infinitely.

## RESET PIC Flow Diagram

Diagram

Description automatically generated

RESET manages all the mission switches, power supply and counting of time from the moment of deployment. In the program flow, we setup variables and then verify which day and if it is the first time the satellite is turning on. This is done to delay the turn on of COMM PIC and the transceiver. After this, we then collect all RESET data which in includes the time data and power supply information (3.3V, UNREG, 5V etc.). The first check is to determine if the time has elapsed to turn on COMM and the transceiver. The second check is to determine if the MAIN PIC is requesting RESET data and to forward if it is. Next, if a burner circuit command was received from main pic, then we activate the antenna deployment burner circuit for 45 seconds. The next check is to determine if main PIC is operating properly. If main pic has not sent communication in at least 1 hour, we will manually reset MAIN PIC. If a command 0x27 was received, we will reset the entire satellite. Similarly, if 24hrs have elapsed, then the reset flag will raise, and we will reset the entire satellite.

## COM PIC Flow Diagram

Diagram

Description automatically generated

COM PIC listens for commands from ground station and executes or forwards as necessary. The first step is the request for the newest CW information from MAIN. Next, we check what is the current CW delay. We then check if a command for rsv data downlink was received. The next series of checks are based on data received from the ground station. If the second byte of the command string is 0x90, we change the data downlink delay. If the byte is 0x11, we prepare data to be downlinked by the GS. If the byte is 0x22, we transfer data between flash memories. If the byte is 0xA0, then the command is designated for MAIN PIC. And finally, if the command is 0x20, we request a CW update from MAIN.

## Mission Boss PIC Flow Diagram

Diagram

Description automatically generated

MISSION BOSS PIC serves as an extension of MAIN PIC. It controls the connections of the UART and SPI lines from the missions to MAIN and the flash memory. The program flow checks which MCU the command is designated for and changes the control pins of the CPLD to allow forwarding the commands to the MCU.