Testing of all Arduino Firmware

Contents:

1. Comms1\_2.h
2. EepromMangager.h
3. Settings.h
4. BMS
5. TempControl

Comms1\_2.h

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No. | Test Name | Success Criteria | Input | Output | Pass or Fail |
| 1 | Echo test for read and write of  Strings -  Bluetooth | Interprets incoming string as strings, takes it apart and converts correctly into opcode and operands.  Takes a character and an array and transmits them together as strings.  Correct Delimiters must be used, commas in the right place. | Phone:  "f27,282,27 " | Phone:  "?f27,282,27!"  Serial Monitor:  "f  27  282  27"  As expected | Pass |
|  |  |  | Phone:  "h284895,2738,0 " | Phone:  "?h284895,2738,0!"  Serial Monitor:  "h  284895  2738  0"  As expected | Pass |
| 2 | Echo test for  Read and Write of Bytes  Bluetooth | Interprets received string as bytes, takes it apart and converts correctly into opcode and operands.  Takes a character and an array and transmits them together as strings.  Correct Delimiters must be used, no commas | Phone:  "a>}a " | Phone:  "?a6212597!"  Serial Monitor:  "?a62,125,97!"  Correct interpretation.  Write failure  After fix  Phone:  "?a>}a!"  As expected | Fail |
|  |  |  | Phone:  "pp. " | Phone:  ""  Serial Monitor:  "Invalid data length  Invalid data length  "  Flaw in sending numbers as bytes. Unable to send numbers 32, 33, 63 as they are the delimiters | Fail |

EepromManager.h

Tested with Settings:

Lower Bound is 511

Upper Bound: 1023

Stamp Size: 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no. | Test Name | Success Criteria | Input | Output | Pass or Fail |
| 3 | Wear Levelling Write | Writes data successfully at position of current iterator | Serial Monitor:  "w489,, " at "itAddress: 515" | Printed entire memory range to Serial monitor:  "515 6  516 0  517 1  518 233"  Data saved as bytes:  233 + 1\*28 = 489  As expected. | pass |
|  |  |  | Serial Monitor:  "w456987,, " at "itAddress: 519" | Printed memory range to Serial monitor:  "519 6  520 6  521 249  522 27"  Data saved as bytes:  27 + 249\*28 +6\*216 = 456987  As expected. | pass |
| 4 | Read | Returns the last saved data correctly | Serial Monitor:  "r,, " when "itAddress: 843" | Serial Monitor:  "lastSavedData: 457194  Read data: 457194"  As expected. | Pass |
| 5 | Iterator after write | Iterator is left at next address to be written | Serial Monitor:  "w8968,, " | Serial Monitor:  "Marker: 5  itAddress: 1015  lastSavedData: 45  Marker: 5  itAddress: 1019  lastSavedData: 8968"  ItAddress has been updated from 1015 to 1019 as expected. | Pass |
| 6 | Iterator after read | Iterator should be left at Address of next write | Serial Monitor:  "r,, " at "itAddress: 999" | Serial Monitor:  "Marker: 7  itAddress: 999  lastSavedData: 6554390  Read data: 6554390  Marker: 7  itAddress: 999  lastSavedData: 6554390"  ItAddress starts and ends at 999 as expected. |  |
| 7 | Is the buffer cyclic with write | Iterator is checked at the beginning of each write and should cycle back to the beginning of range after it goes out of effective range | Serial Monitor:  "w453,, "  At "itAddress: 1023"  (out of range) | "Marker: 5  itAddress: 1023  lastSavedData: 79  Was: 1023  Now: 511  Incremented marker  Marker: 6  itAddress: 515  lastSavedData: 453"  As expected. | pass |
| 8 | Is the buffer cyclic with read | Reads last EEPROM stamp correctly when the next write should be at the start of the range. | Serial Monitor:  "r,, " at "itAddress: 511"  Memory that should be read  "1019 14  1020 100  1021 7  1022 27"  Should return 6555419 | "Marker: 0  itAddress: 511  lastSavedData: 65535  Read data: 0  Marker: 0  itAddress: 1022  lastSavedData: 0"  Program reads "0" as from EEPROM  After fix: cast to ulong and changed to |= operator, incremented itAddress:  "Marker: 0  itAddress: 511  lastSavedData: 65535  Read data: 6555419  Marker: 0  itAddress: 511  lastSavedData: 6555419"  Reads "6555419" as expected | pass |
| 9 | Correct set up on start up  Begin() | ItAddress should be set to that of the next address to be written to. Last saved data should also be set to the last value | Serial Monitor:  "r,, "  "711 13  712 100  713 6  714 78  715 13  716 100  717 6  718 79  719 12  720 100  721 5  722 208  723 12  724 100  725 5  726 209"  Here the marker is 13 (at address 711, 715). The last next stamp to be overwritten is the first address with a 12 (address 719) | Serial Monitor:  "Marker: 13  itAddress: 719  lastSavedData: 6555215  Read data: 6555215  Marker: 13  itAddress: 719  lastSavedData: 6555215"  ItAddress is initialised to 719 after begin() is called, lastSavedData is 6555215 = 79 + 6\*28 + 100\*216 as expected | pass |

Settings.h

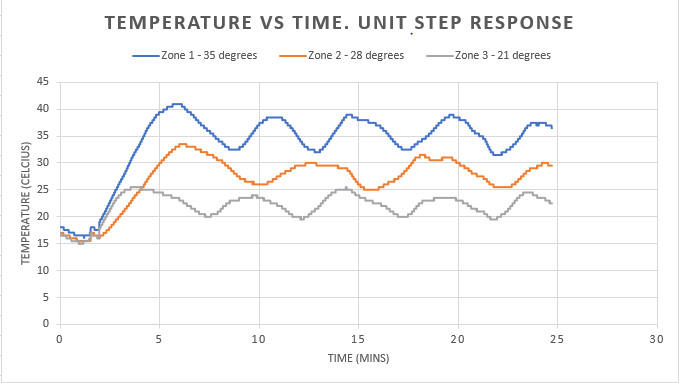
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no. | Test Name | Success Criteria | Input | Output | Pass or Fail |
| 10 | Correct storage of settings | Addresses in EEPROM should be correctly written to. storeSettingsOfMode() | Serial Monitor:  "n210,132,67 " | Printed EEPROM memory range to Serial Monitor:  "1 210  2 132  3 67"  The correct values have been saved in the correct addresses as expected | pass |
|  |  |  | Serial Monitor:  "p12,1,0 " | Printed EEPROM memory range to Serial Monitor:  "7 12  8 1  9 0"  Once again correct values have been stored in the correct position. | pass |
| 11 | Correct retrieval of settings | Addresses in EEPROM should be correctly read. getSettingsOfMode() | Serial Monitor:  "a,, "  Printed EEPROM memory range:  "  Address Stored Value  0 1  1 210  2 132  3 67  4 3  5 100  6 0  7 12  8 1  9 0" | Serial Monitor:  "?a210,132,67!"  Settings returned for mode 1 were 210, 132, 67 as expected. | pass |
|  |  |  | Serial Monitor:  "b,, "  EEPROM memory remained unchanged | Serial Monitor:  "?b3,100,0!"  Settings returned for mode 1 were 3, 100 and 0 as expected | pass |
| 12 | Update current settings | Settings read from EEPROM and then written to current settings correctly writeCurrentMode()  Number at the root address should los update to remember the last | Serial Monitor:  "q2,, "  Printed EEPROM memory range on settings:  "Print range:  Address Stored Value  0 1  1 89  2 73  3 253  4 43  5 126  6 78  7 255  8 34  9 75" | Serial Monitor:  "Current Mode: 2  Zone 1: 43  Zone 2: 126  Zone 3: 78"  "Print range:  Address Stored Value  0 2  1 89  2 73  3 253  4 43  5 126  6 78  7 255  8 34  9 75"  Current Settings updated and root address updated to 2 as expected | pass |
|  |  |  | Serial Monitor:  "q3,, "  EEPROM Settings remain unchanged | Serial Monitor:  "Current Mode: 3  Zone 1: 255  Zone 2: 34  Zone 3: 75"  "Print range:  Address Stored Value  0 3"  Settings correctly updated and value in root address updated to 3 as expected | pass |
| 13 | Cycle Mode | Mode correctly goes to next save | Serial Monitor:  "y,, " with current mode at 1. EEPROM Settings remain unchanged | Serial Monitor:  "Current Mode: 1  Zone 1: 89  Zone 2: 73  Zone 3: 253    CycleMode  Current Mode: 2  Zone 1: 43  Zone 2: 126  Zone 3: 78"  Current mode went from mode 1 to 2 correctly as expected | pass |
|  |  | Mode correctly loops round to mode 1 | Serial Monitor:  "y,, " with current mode at 3. EEPROM Settings remain unchanged | Serial Monitor:  "Current Mode: 3  Zone 1: 255  Zone 2: 34  Zone 3: 75    CycleMode  Current Mode: 1  Zone 1: 89  Zone 2: 73  Zone 3: 253"  Current mode and settings went from mode 3 to 1 correctly as expected | pass |

Battery Management System

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no. | Test Name | Success Criteria | Input | Output | Pass Fail |
| 14 | SampleSensors() | Sensors give smooth and beleivable readings I.e. voltage is between 3.5 and 4.2, current is between –5A and 5A | Connected Heaters up to the board, connected batteries. Set the Switches to fully on | Corrected small error with vtotal initialisation.  Changed data type of v total and itotal to unsigned int to avoid possibility of overflow. When adding together 30 numbers up to 1023.  Serial Monitor:  Figures for cell volts:  "5.00, -4.50, -0.50  5.00, -4.50, -0.50  5.00, -4.50, -0.50  5.00, -4.50, -0.50" Negative numbers not expected.  Switched the cell pins around:  "0.15, 1.22, -1.37  0.15, 1.23, -1.38  0.15, 1.22, -1.37  0.15, 1.22, -1.37" We also get negative numbers.  After hardware fix  Correct values were given as expected | pass |
| 15 | InitialiseArrays() | Rest of the sensor sensing should work fine | Turned belt on | Corrected function parameters  Readings plausible as expected | pass |
| 16 | BMS() | Provides overvoltage protection | Batteries charged till at least 1 cell approaches danger threshold. | "Mosfet 1 closed  .  .  .  .  Mosfet 1 opened"  The mosfets are discharged approriately as expected  Resistors get a bit hot | pass |
| 17 | Led.lightUp()  colour | Green for capacity consumed < 66% battery Capacity for "fully charged, Red for capacity consumed is >66% Battery Capacity for "low battery" and yellow for when it is charging | Set battery capacity to 700. 500, 300 | Colour is yellow at 700, green at 500 and red at 300. Colours are wrong. Condition for yellow is not right. I may have accidentally swapped green and red pins  After fix  When capacity consumed is 700 and current is +ve, colour is red. Capacity consumed is 300 or 500 and current is +ve, colour is green. When current is –ve, colour is yellow. As expected | pass |
| 18 | Led.blink()  number | The LED is set to blink 1 time to indicate mode 1, 2 times to indicate mode 2 and 3 times to indicate mode 3. | Set mode to 1,2 and 3 | In mode 1, Led blinks once. In mode2, Led blinks twice. In mode 3, Led blinks three times as expected | pass |
| 19 | GetMpptStatus() | CANNOT TEST |  |  |  |
| 20 | UpdateCapacityConsumed() | Linear Test. Constant current will be set, if successful, capacity consumed will be within 10% of analytical calculation of charge. | Set current = 2, run for 10 seconds.  With the equation in the function which gives consumed capacity in Amp seconds, should get 0, 20As, 40As, 60s... | Serial Monitor:  "Capacity consumed: 0  Capacity consumed: 0  Capacity consumed: 0  Capacity consumed: 0  Capacity consumed: 0  Capacity consumed: 0" program returns 0s due to rounding errors  After fix -changed data type from int to float.  "Capacity consumed: 0.42  Capacity consumed: 11.40  Capacity consumed: 20.04  Capacity consumed: 29.50"  Values half of expected  -corrected timing as well  "Capacity consumed: 0.00  Capacity consumed: 18.95  Capacity consumed: 36.57  Capacity consumed: 54.55  Capacity consumed: 72.71  Capacity consumed: 91.65  "  Numbers are better, there is still drift | Partial pass |

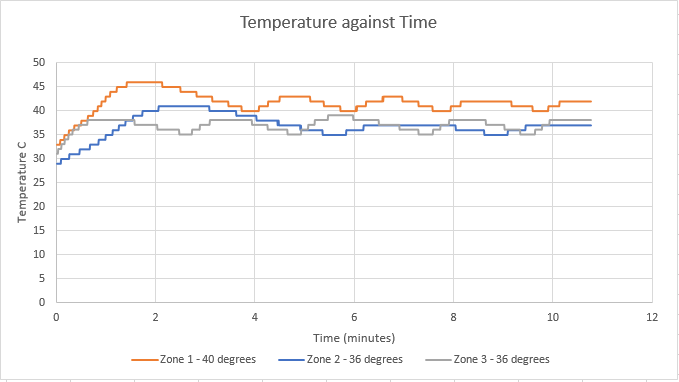
TempControl

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test no. | Test Name | Success Criteria | Input | Output | Pass Fail |
| 21 | Temperature Sense | Valid temperatures are mapped to 0 –65536 returned and change when expected | Temperature sensors are attached to correct pins | "Temp C: 33704Temp C: 32768Temp C: 33704  Temp C: 33704Temp C: 32768Temp C: 33704  Temp C: 33704Temp C: 32768Temp C: 33704  Temp C: 33704Temp C: 32768Temp C: 33704  Temp C: 33704Temp C: 32768Temp C: 33704  Temp C: 33704Temp C: 32768Temp C: 33704"  With the mapping that I have used, this corresponds to a temperaure of 26°C as expected | pass |
| 22 | Control loop | Loop remains around the set point | Zone 1, 2, 3 set to 21, 28, 35 degrees celcius |  | Partial pass |



As you can see, temperatures oscillate around set points as expected, but the amplitude of oscillation is unacceptable.

Temperature sensors are placed closer to the heaters to reduce the hysteresis. Zone 1, 2, 3 set to 40, 36, 36 to emulate a possible custom setting



Hysteresis is greatly reduced - pass