| Revision History | | | |
| --- | --- | --- | --- |
| Date | Version | By | Description of Change |
| 21 Aug, 2022 | 0.01 | Kunming Yang | First version |
| 29 Aug, 2022 | 0.02 | Jerry Wang | Define some Req |
| 09 Sep,2022 | 0.03 | Kunming Yang | Update as review |
| 19 Sep,2022 | 0.04 | Kunming Yang | Update as review |
| 21 Sep, 2022 | 0.05 | Jerry Wang | Define Weld Result and Weld Signature work flow |
| 22 Sep, 2022 | 0.06 | Kunming Yang | Update as review |
| 29 Sep, 2022 | 0.07 | Kunming Yang | Change type of some Table ID to long long |
| 09 Oct, 2022 | 0.08 | Kunming Yang | Create all 19 tables |
|  |  |  |  |

# Schedule

|  |  |
| --- | --- |
| Day | Sprint 1st |
| 1 | Database concurrent mechanism |
|  | 1st Week |
| 6 | Database architecture review |
|  | 2nd Week |
| 11 | Database initialization workflow review |
| 14 | Database Interface & message queue definition |
| 15 | Recipe, WeldResult, Weld Signature |
|  | 3rd Week |
| 21 | Code Review |
|  | Sprint 2nd |
|  | 1st Week |
| 25 | Performance testing |
|  | 2nd Week |
| 1 | Test Report Review |
| 2 | Database daemon task |
|  | 3rd Week |
| 8 | Code Review |

# DataBase

## General requirements

1. The database file should be named as sample\_l20\_base.db
2. The database file should be stored in /mmc1
3. The database should include 19 tables, with names
   1. WeldRecipe
   2. WeldResult
   3. WeldResultSignature
   4. AlarmLog
   5. SystemConfiguration
   6. EventLog
   7. MaintenanceCounter
   8. UserProfile
   9. PrivilegeConfiguration
   10. PrivilegeLevelName
   11. Sequence
   12. SequencePreset
   13. LastOperate
   14. Communication
   15. GlobalSetting
   16. HeightCalibration
   17. MaintenanceLog
   18. TeachModeConfiguration
   19. DBVersion
4. To be supplemented

## Table WeldRecipe

Max count 1000

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | WeldRecipe | | | | | | |
| Primary Key | | ID | | | | | | |
| Other fields | | RecipeName, DateTime, UserID, PresetPicPath, IsVerified, Amplitude, Width, WeldPressure, TriggerPresure, TimePlus, TimeMinus, PeakPowerPlus, PeakPowerMinus, TriggerHeightPlus, TriggerHeightMinus, WeldHeightPlus, WeldHeightMinus, WeldMode, ModeValue, PreBurst, HoldTime, SqueezeTime, AfterBurstDelay, AfterBurstDuration, AfterBurstAmplitude, WeldHeight, MeasuredHeight, StepWeldMode, EnergyToStep, TimeToStep, PowerToStep | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | ID | INTEGER | Y | Y | Y |  | INTEGER | 1 |
|  | RecipeName | VARCHAR(100) | Y | N | Y |  | TEXT | “aaa” |
|  | DateTime | TEXT | Y | N | Y |  | TEXT | 2022-01-31 01:02:03 |
|  | UserID | INTEGER | Y | N | N |  | INTEGER | 1 |
|  | PresetPicPath | VARCHAR | N | N | N |  | TEXT | "D: \picture” |
|  | IsValidate | BOOLEAN | Y | N | N | 0 | INTEGER | 0(false) |
|  | Amplitude | INTEGER | Y | N | N | 18 | INTEGER | 18(um) |
|  | Width | INTEGER | Y | N | N | 2000 | INTEGER | 2000(micrometer) |
|  | WeldPressure | INTEGER | Y | N | N | 20 | INTEGER | 20(psi) \* 1000 |
|  | TriggerPressure | INTEGER | Y | N | N | 20 | INTEGER | 20(psi) \* 1000 |
|  | TimePlus | INTEGER | Y | N | N | 5000 | INTEGER | 5000(ms) |
|  | TimeMinus | INTEGER | Y | N | N | 0000 | INTEGER | 0000(ms) |
|  | PeakPowerPlus | INTEGER | Y | N | N | 4800 | INTEGER | 4800(W) |
|  | PeakPowerMinus | INTEGER | Y | N | N | 0 | INTEGER | 0(W) |
|  | TriggerHeightPlus | INTEGER | Y | N | N | 15000 | INTEGER | 15000(micrometer) |
|  | TriggerHeightMinus | INTEGER | Y | N | N | 0000 | INTEGER | 0000(micrometer) |
|  | WeldHeightPlus | INTEGER | Y | N | N | 15000 | INTEGER | 15000(micrometer) |
|  | WeldHeightMinus | INTEGER | Y | N | N | 0000 | INTEGER | 0000(micrometer) |
|  | WeldMode | ENUM | Y | N | N | 0 | INTEGER | 0(Energy) |
|  | ModeValue | INTEGER | Y | N | N | 0 | INTEGER | 100(J) |
|  | PreBurst | INTEGER | Y | N | N | 0000 | INTEGER | 0000(ms) |
|  | HoldTime | INTEGER | Y | N | N | 0000 | INTEGER | 0000(ms) |
|  | SqueezeTime | INTEGER | Y | N | N | 0000 | INTEGER | 0000(ms) |
|  | AfterBurstDelay | INTEGER | Y | N | N | 1000 | INTEGER | 1000(ms) |
|  | AfterBurstDuration | INTEGER | Y | N | N | 0000 | INTEGER | 0000(ms) |
|  | AfterBurstAmplitude | INTEGER | Y | N | N | 0000 | INTEGER | 0000(ms) |
|  | WeldHeight | INTEGER | Y | N | N | 50 | INTEGER | 50(micrometer) |
|  | MeasuredHeight | INTEGER | Y | N | N | 30 | INTEGER | 30(micrometer) |
|  | StepWeldMode | ENUM | N | N | N | -1 | INTEGER | -1(disable) |
|  | EnergyToStep | JSON | N | N | N | 0 | BLOB | {"0": [1, 4, 0, 0, 7], "1": [2, 5, 0, 0, 8], "2": [3, 6, 0, 0, 9]} |
|  | TimeToStep | JSON | N | N | N | 0 | BLOB | {"0": [1, 4, 0, 0, 7], "1": [2, 5, 0, 0, 8], "2": [3, 6, 0, 0, 9]} |
|  | PowerToStep | JSON | N | N | N | 0 | BLOB | {"0": [1, 4, 0, 0, 7], "1": [2, 5, 0, 0, 8], "2": [3, 6, 0, 0, 9]} |
| Sqlite scripts | | 1. --Create the table   CREATE TABLE "WeldRecipe" (  "ID" INTEGER NOT NULL UNIQUE,  "RecipetName" TEXT NOT NULL UNIQUE,  "DateTime" TEXT NOT NULL,  "UserID" INTEGER,  "PresetPicPath" TEXT,  "IsVerified" INTEGER NOT NULL DEFAULT 0,  "Amplitude" INTEGER NOT NULL DEFAULT 18,  "Width" INTEGER NOT NULL DEFAULT 2000,  "WeldPressure" INTEGER NOT NULL DEFAULT 20000,  "TriggerPressure" INTEGER NOT NULL DEFAULT 20000,  "TimePlus" INTEGER NOT NULL DEFAULT 5000,  "TimeMinus" INTEGER NOT NULL DEFAULT 0,  "PowerPlus" INTEGER NOT NULL DEFAULT 4800,  "PowerMinus" INTEGER NOT NULL DEFAULT 0,  "TriggerHeightPlus" INTEGER NOT NULL DEFAULT 15000,  "TriggerHeightMinus" INTEGER NOT NULL DEFAULT 0,  "WeldHeightPlus" INTEGER NOT NULL DEFAULT 15000,  "WeldHeightMinus" INTEGER NOT NULL DEFAULT 0,  "WeldMode" INTEGER NOT NULL DEFAULT 0,  “ModeValue” INTEGER NOT NULL DEFAULT 100,  "PreBurst" INTEGER NOT NULL DEFAULT 0,  "HoldTime" INTEGER NOT NULL DEFAULT 0,  "SqueezeTime" INTEGER NOT NULL DEFAULT 0,  "AfterBurstDelay" INTEGER NOT NULL DEFAULT 1000,  "AfterBurstDuration" INTEGER NOT NULL DEFAULT 0,  "AfterBurstAmplitude" INTEGER NOT NULL DEFAULT 0,  "WeldHeight" INTEGER NOT NULL DEFAULT 0,  "MeasuredHeight" INTEGER NOT NULL DEFAULT 0,  "StepWeldMode" INTEGER DEFAULT 0,  "EnergyToStep" BLOB DEFAULT 0,  "TimeToStep" BLOB DEFAULT 0,  "PowerToStep" BLOB DEFAULT 0,  FOREIGN KEY("UserID") REFERENCES "UserProfiles"("ID"),  PRIMARY KEY("ID" AUTOINCREMENT)  );   1. --Insert a record   INSERT INTO WeldRecipe (PresetName,CreatedDate,UserID,PresetPicPath,Verified,..) VALUE ("aaa","2021-11-11 12:12:12 666",1,"D:\yz\Other\picture",0,..) | | | | | | |
| Tips | | Json format:  {Order0: [energy0, amplitude0],  Order1: [energy1, amplitude1],  Order2: [energy2, amplitude2]}  for example, {“0”: [0,0], “1”: [1, 100]} | | | | | | |
| Notes: | | The table operands should include INSERT for a new record, DELETE oldest records using quantity, UPDATE the record following ID, QUERY ID, DateTime and RecipeName using quantity; QUERY a record using ID. | | | | | | |

## Table WeldResult

Max count 1000000

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | WeldResult | | | | | | |
| Primary Key | | ID | | | | | | |
| Other fields | | UserID, DateTime, SequenceID, RecipeID, WeldEnergy, TriggerPressure, WeldPressure, WeldAmplitude, WeldTime, WeldPeakPower, TriggerHeight, WeldHeight,AlarmFlags, CycleCounter | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | ID | INTEGER | Y | Y | Y |  | INTEGER | 4294967298 (64bit) |
|  | PartID | TEXT[50] | Y | N | N |  | TEXT | 1 |
|  | DateTime | TEXT | Y | N | N |  | TEXT |  |
|  | RecipeID | INTEGER | N | N | N |  | INTEGER | 1 |
|  | WeldEnergy | INTEGER | Y | N | N |  | INTEGER | 100 |
|  | TriggerPressure | INTEGER | Y | N | N |  | INTEGER | 20.0(psi) \* 1000 |
|  | WeldPressure | INTEGER | Y | N | N |  | INTEGER | 20.0(psi) \* 1000 |
|  | WeldAmplitude | INTEGER | Y | N | N |  | INTEGER | 18(um) |
|  | WeldTime | INTEGER | Y | N | N |  | INTEGER | 1500(ms) |
|  | WeldPeakPower | INTEGER | Y | N | N |  | INTEGER | 100(W) |
|  | TriggerHeight | INTEGER | Y | N | N |  | INTEGER | 4250(micrometre) |
|  | WeldHeight | INTEGER | Y | N | N |  | INTEGER | 4250(micrometre) |
|  | AlarmFlag | INTEGER | Y | N | N |  | INTEGER | 0 |
|  | SequenceID | INTEGER | N | N | N |  | INTEGER | 1 |
|  | CycleCounter | INTEGER | N | N | N |  | INTEGER | 1000 |
| Sqlite scripts | | 1. --Create the table   CREATE TABLE "WeldResult" (  "ID" INTEGER NOT NULL UNIQUE,  "PartID" TEXT NOT NULL,  "DateTime" TEXT NOT NULL,  "SequenceID" INTEGER,  "RecipeID" INTEGER,  "WeldEnergy" INTEGER NOT NULL,  "TriggerPressure" INTEGER NOT NULL,  "WeldPressure" INTEGER NOT NULL,  "WeldAmplitude" INTEGER NOT NULL,  "WeldTime" INTEGER NOT NULL,  "WeldPeakPower" INTEGER NOT NULL,  "TriggerHeight" INTEGER NOT NULL,  "WeldHeight" INTEGER NOT NULL,  "AlarmFlag " INTEGER,  "CycleCounter” INTEGER,  PRIMARY KEY("ID" AUTOINCREMENT)  );   1. --Insert a record   INSERT INTO WeldResult (PartID, DateTime, SequenceID, RecipeID, WeldEnergy, TriggerPressure, WeldPressure, WeldAmplitude, WeldTime, WeldPeakPower, TriggerHeight, WeldHeight, AlarmFlag, CycleCounter) VALUES (“1”, “2021-11-11 12:12:12”,1,1,100,20000,20000,18,1500,100,100,4250,4250,0,200) | | | | | | |
| Tips | |  | | | | | | |
| Notes | | Should same with Alarm Log. | | | | | | |

## Table WeldResultSignature

Max count 5000

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | WeldResultSignature | | | | | | |
| Primary Key | | ID | | | | | | |
| Other fields | | WeldResultID, WeldGraph | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | ID | INTEGER | Y | Y | Y |  | INTEGER | 4294967298 (64bit) |
|  | WeldResultID | INTEGER | Y | N | Y |  | INTEGER | 4294967298 (64bit) |
|  | WeldGraph | JSON | N | N | N |  | BLOB | {"0": [1, 2, 34, 5, 6, 6],  "1": [1.5, 2, 1, 1.9, 2.0]} |
| Sqlite script | | 1. --Create the table   CREATE TABLE "WeldResultGraph" (  "ID" INTEGER NOT NULL UNIQUE,  "WeldResultID" INTEGER NOT NULL UNIQUE,  "WeldGraph" TEXT NOT NULL,  FOREIGN KEY("WeldResultID") REFERENCES "WeldResult"("ID") ON DELETE CASCADE,  PRIMARY KEY("ID" AUTOINCREMENT)  );   1. --Insert a record   INSERT INTO WeldResultGraph (WeldResultID, WeldGraph) VALUES (1," {"0": [1, 2, 34, 5, 6, 6],"1": [1.5, 2, 1, 1.9, 2.0]}") | | | | | | |
| Tips | | WeldGraph data format: {“WeldGraphType”: curve data}  The WeldGraph Type is a Enum type and its detail is shown in the chapter5.3. | | | | | | |
| Notes | | Should include INSERT and QUERY record following “WeldResultID”. | | | | | | |

## Table AlarmLog

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | AlarmLog | | | | | | |
| Primary Key | | ID | | | | | | |
| Sort other fields | | AlarmMsg, CreateDate, AlarmType, PresetID, UserID, IsReset, WeldResultID | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | ID | INTEGER | Y | Y | Y |  | INTEGER | 1 |
|  | AlarmMsg | VARCHAR(200) | Y | N | N |  | TEXT | “Exceed the height value” |
|  | CreateDate | DATETIME | Y | N | N |  | TEXT | 12-12-2022 12:12:21 345 |
|  | AlarmType | ENUM | Y | N | N |  | INTEGER | 1 |
|  | PresetID | INTEGER | N | N | N |  | INTEGER | 1 |
|  | UserID | INTEGER | Y | N | N |  | INTEGER | 1 |
|  | IsReseted | INTEGER | Y | N | N | 0 | INTEGER | 0 |
|  | WeldResultID | INTEGER | N | N | N |  | INTEGER | 1 |
| Sqlite script | | 1. --Create table   CREATE TABLE "AlarmLog" (  "ID" INTEGER NOT NULL UNIQUE,  "AlarmMsg" TEXT NOT NULL,  "CreateDate" TEXT NOT NULL,  "AlarmType" INTEGER NOT NULL,  "PresetID" INTEGER,  "UserID" INTEGER NOT NULL,  "IsReset" INTEGER NOT NULL,  "WeldResultID" INTEGER NOT NULL UNIQUE,  FOREIGN KEY("WeldResultID") REFERENCES "WeldResult"("ID") ON DELETE CASCADE,  PRIMARY KEY ("ID" AUTOINCREMENT)  );   1. --Insert a record   (AlarmMsg,CreateDate,AlarmType, PresetID, UserID,IsReset,WeldResultID) VALUES ("exceed the max value",”2021-11-26 12:12:12 999”,1,1,1,0,999) | | | | | | |
| Tips | |  | | | | | | |
| Notes: | | The table operands should include INSERT for a whole record, DELETE oldest records using quantity, UPDATE “IsReset” field following ID, QUERY records using “IsReset”, Date and quantity. | | | | | | |

## Table SystemConfiguration

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | SystemConfiguration | | | | | | |
| Primary Key | | -- | | | | | | |
| Other fields | | PressureUnit, HeightUnit, Language, AuthorityCheck | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Values |
|  | PressureUnit | ENUM | Y | N | Y | 0 | INTERGER | 0(psi) |
|  | HeightUnit | ENUM | Y | N | Y | 0 | INTERGER | 0(mm) |
|  | Language | ENUM | Y | N | Y | 0 | INTERGER | 0(English) |
|  | AuthorityCheck | BOOLEAN | Y | N | Y | 0 | INTERGER | 0(off) |
| Sqlite script | | 1. --Create the table   CREATE TABLE "SystemConfiguration" (  "PressureUnit" INTEGER NOT NULL UNIQUE,  "HeightUnit " INTEGER NOT NULL UNIQUE,  "Language" INTEGER NOT NULL UNIQUE,  "AuthorityCheck" INTEGER NOT NULL UNIQUE  );   1. --Insert a record   INSERT INTO SystemConfiguration VALUES (…)   1. --Update the record   UPDATE SystemConfiguration SET Language =1 | | | | | | |
| Tips | |  | | | | | | |
| Notes: | | Please ignore it firstly. | | | | | | |

## Table EventLog

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | EventLog | | | | | | |
| Primary Key | | ID | | | | | | |
| Other fields | | Datetime, EventType, EventName, UserID, Old, New, Comment | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | ID | INTEGER | Y | Y | Y |  | INTEGER | 1 |
|  | Datetime | TEXT | Y | N | N |  | TEXT | 12-12-2021 12:12:12 |
|  | EventType | ENUM | Y | N | N |  | INTEGER | 1 |
|  | EventName | TEXT | Y | N | N |  | TEXT | “Modify Parameter” |
|  | UserID | INTEGER | Y | N | N |  | INTEGER | 1 |
|  | Old | TEXT | Y | N | N |  | TEXT | 5 |
|  | New | TEXT | Y | N | N |  | TEXT | 4 |
|  | Comment | VARCHAR(200) | N | N | N |  | TEXT | “Modify the param” |
| Sqlite script | | 1. --Create the table   CREATE TABLE "EventLog" (  "ID" INTEGER NOT NULL UNIQUE,  "Datetime" TEXT NOT NULL,  "EventType" INTEGER NOT NULL,  "EventName" TEXT NOT NULL,  "UserID" INTEGER NOT NULL,  "Old" TEXT NOT NULL,  "New" TEXT NOT NULL,  "Comment" TEXT,  PRIMARY KEY ("ID" AUTOINCREMENT)  );   1. --Insert a record   INSERT INTO EventLog (Datetime,EventType,EventName, UserID,Old,New,Comment) VALUES ("2021-11-11 12:12:12 666", 1, "Modify parameter", 1, "old", "New", "Changed the parameter") | | | | | | |
| Tips | |  | | | | | | |
| Notes | | Same with Alarm Table, please ignore it firstly. | | | | | | |

## Table MaintenanceCounter

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | MaintenanceCounter | | | | | | |
| Primary Key | | -- | | | | | | |
| Other fields | | HornCurrentCounter,HornEnergyCurrentCounter,HornCounterLimit,HornEnergyLimit,HornStartDate,HornCounter100Alarm,HornEnergy100Alarm,AnvilCurrentCounter,AnvilEnergyCurrentCounter,AnvilCounterLimit,AnvilEnergyLimit,AnvilStartDate,AnvilCounter100Alarm,AnvilEnergy100Alarm,ConverterCurrentCounter,ConverterEnergyCurrentCounter,ConverterCounterLimit,ConverterEnergyLimit,ConverterStartDate,ConverterCounter100Alarm,ConverterEnergy100Alarm,GeneratorCurrentCounter,GeneratorEnergyCurrentCounter,GeneratorCounterLimit,GeneratorEnergyLimit,GeneratorStartDate,GeneratorCounter100Alarm,GeneratorEnergy100Alarm,ActuatorCounter,ActuatorStartDate | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | HornCurrentCounter | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | HornEnergySum | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | HornCounterLimit | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | HornEnergyLimit | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | HornStartDate | DATETIME | Y | N | Y | --/--/-- | TEXT | 12-12-2021 12:12:12 |
|  | HornCounter100AlarmOption | BOOL | Y | N | Y | 0 | INTEGER | 1(on) |
|  | HornEnergy100AlarmOption | BOOL | Y | N | Y | 0 | INTEGER | 1(on) |
|  | AnvilCurrentCounter | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | AnvilEnergyCurrentSum | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | AnvilCounterLimit | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | AnvilEnergyLimit | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | AnvilStartDate | DATETIME | Y | N | Y | --/--/-- | TEXT | 12-12-2021 12:12:12 |
|  | AnvilCounter100AlarmOption | BOOL | Y | N | Y | 0 | INTEGER | 1(on) |
|  | AnvilEnergy100AlarmOption | BOOL | Y | N | Y | 0 | INTEGER | 1(on) |
|  | ConverterCurrentCounter | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | ConverterEnergySum | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | ConverterCounterLimit | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | ConverterEnergyLimit | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | ConverterStartDate | DATETIME | Y | N | Y | --/--/-- | TEXT | 12-12-2021 12:12:12 |
|  | ConverterCounter100AlarmOption | BOOL | Y | N | Y | 0 | INTEGER | 1(on) |
|  | ConverterEnergy100AlarmOption | BOOL | Y | N | Y | 0 | INTEGER | 1(on) |
|  | GeneratorCurrentCounter | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | GeneratorEnergySum | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | GeneratorCounterLimit | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | GeneratorEnergyLimit | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | GeneratorStartDate | DATETIME | Y | N | Y | --/--/-- | TEXT | 12-12-2021 12:12:12 |
|  | GeneratorCounter100AlarmOption | BOOL | Y | N | Y | 0 | INTEGER | 1 |
|  | GeneratorEnergy100AlarmOption | BOOL | Y | N | Y | 0 | INTEGER | 1 |
|  | ActuatorCounter | INTEGER | Y | N | Y | 0 | INTEGER | 0 |
|  | ActuatorStartDate | DATETIME | Y | N | Y | --/--/-- | TEXT | 12-12-2021 12:12:12 |
| Sqlite scripts | | 1. --Create the table   CREATE TABLE "MaintenanceCounter" (  "HornCurrentCounter" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "HornEnergySum" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "HornCounterLimit" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "HornEnergyLimit" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "HornStartDate" TEXT NOT NULL UNIQUE,  "HornCounter100AlarmOption" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "HornEnergy100AlarmOption" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "AnvilCurrentCounter" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "AnvilEnergyCurrentSum" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "AnvilCounterLimit" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "AnvilEnergyLimit" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "AnvilStartDate" TEXT NOT NULL UNIQUE,  "AnvilCounter100AlarmOption" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "AnvilEnergy100AlarmOption" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "ConverterCurrentCounter" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "ConverterEnergysum" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "ConverterCounterLimit" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "ConverterEnergyLimit" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "ConverterStartDate" TEXT NOT NULL UNIQUE,  "ConverterCounter100AlarmOption" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "ConverterEnergy100AlarmOption" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "GeneratorCurrentCounter" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "GeneratorEnergySum" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "GeneratorCounterLimit" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "GeneratorEnergyLimit" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "GeneratorStartDate" TEXT NOT NULL UNIQUE,  "GeneratorCounter100AlarmOption" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "GeneratorEnergy100AlarmOption" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "ActuatorCounter" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "ActuatorStartDate" TEXT NOT NULL UNIQUE  );   1. --Insert a record   INSERT INTO MaintenanceCounter VALUES (?,?,…) | | | | | | |
| Tips | |  | | | | | | |
| Notes | | Please ignore it. | | | | | | |

## Table UserProfiles

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | UserProfiles | | | | | | |
| Primary Key | | ID | | | | | | |
| Other fields | | UserName,CreateDate,WhoCreateNewID,Password,PermissionLevel | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Values |
|  | ID | INTEGER | Y | Y | Y |  | INTEGER | 1 |
|  | UserName | VARCHAR(100) | Y | N | Y |  | TEXT | “aaa” |
|  | CreatedDate | DATETIME | Y | N | N |  | TEXT | 12-12-2021 12:12:12 |
|  | WhoCreatedNewID | INTEGER | Y | N | N |  | INTEGER | 1 |
|  | Password | TEXT | Y | N | N |  | TEXT | “0000” |
|  | PermissionLevel | ENUM | Y | N | N |  | INTEGER | 1 |
| Sqlite scripts | | 1. --Create the table   CREATE TABLE "UserProfiles" (  "ID" INTEGER NOT NULL UNIQUE,  "UserName" TEXT NOT NULL UNIQUE,  "CreateDate" TEXT NOT NULL,  "WhoCreateNewID" INTEGER NOT NULL,  "Password" TEXT NOT NULL,  "PermissionLevel" INTEGER NOT NULL,  PRIMARY KEY("ID" AUTOINCREMENT)  );   1. --Insert a record   INSERT INTO UserProfiles (UserName,CreateDate,WhoCreateNewID,Password,PermissionLevel) VALUES ("Def","9999","ADMIN","123456",1) | | | | | | |
| Tips | |  | | | | | | |
| Notes | | Please ignore it. | | | | | | |

## Table PrivilegeConfiguration

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table names | | PrivilegeConfiguration | | | | | | |
| Primary Key | | ID | | | | | | |
| Other fields | | Level1, Level2, Level3, Level4 | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | ID | INTEGER | Y | N | Y |  | INTEGER | 1 |
|  | Level1 | BOOLEAN | Y | N | N | 0 | INTEGER | 0 |
|  | Level2 | BOOLEAN | Y | N | N | 0 | INTEGER | 0 |
|  | Level3 | BOOLEAN | Y | N | N | 0 | INTEGER | 0 |
|  | Level4 | BOOLEAN | Y | N | N | 0 | INTEGER | 0 |
|  | ScreenName | TEXT | Y | N | Y |  | TEXT | “Create Splice” |
| Sqlite scripts | | 1. --Create the table   CREATE TABLE "PrivilegeConfiguration" (  "ID" INTEGER NOT NULL,  "Level1" INTEGER NOT NULL DEFAULT 0,  "Level2" INTEGER NOT NULL DEFAULT 0,  "Level3" INTEGER NOT NULL DEFAULT 0,  "Level4" INTEGER NOT NULL DEFAULT 0,  "ScreenName" TEXT NOT NULL UNIQUE,  PRIMARY KEY("ID")  ); | | | | | | |
| Tips | | The ID means the screen number and the details is shown below table. There needs to insert 20 records when create the table. | | | | | | |
| Notes: | | Please ignore it | | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Screen** | **ID** | **Screen** | **ID** | **Screen** | **ID** | **Screen** |
| 1 | Create Splice | 2 | Create Sequence | 3 | Edit Splice | 4 | Edit Sequence |
| 5 | Operate splice | 6 | Operate Sequence | 7 | Test | 8 | Teach Mode |
| 9 | Calibration | 10 | Advanced Maintenance | 11 | Maintenance Counter | 12 | Maintenance Log |
| 13 | Weld Result History | 14 | Statistical Trend | 15 | Alarm/Error Log | 16 | Library |
| 17 | Version Information | 18 | Permission Setting | 19 | Data Communication | 20 | Operator Library |

## Table PrivileveLevelName

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | PrivileveLevelName | | | | | | |
| Primary Key | | -- | | | | | | |
| Other fields | | Level1, Level2, Level3, Level4 | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | Level1 | TEXT | Y | N | N | Level1 | TEXT | TECHNICAL |
|  | Level2 | TEXT | Y | N | N | Level2 | TEXT | OPERATE |
|  | Level3 | TEXT | Y | N | N | Level3 | TEXT | EXECUTE |
|  | Level4 | TEXT | Y | N | N | Level4 | TEXT | Level4 |
| Sqlite Scripts | | 1. --Create the table   CREATE TABLE "PrivileveLevelName" (  "Level1" TEXT NOT NULL DEFAULT 'Level1',  "Level2" TEXT NOT NULL DEFAULT 'Level2',  "Level3" TEXT NOT NULL DEFAULT 'Level3',  "Level4" TEXT NOT NULL DEFAULT 'Level4'  ); | | | | | | |
| Tips | |  | | | | | | |
| Notes | | Please ignore it | | | | | | |

## Table Sequence

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table names | | Sequence | | | | | | |
| Primary Key | | ID | | | | | | |
| Other fields | | SequenceName, CreatedDate, UserID | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | ID | INTEGER | Y | Y | Y |  | INTEGER | 1 |
|  | SequenceName | VARCHAR (100) | Y | N | Y |  | TEXT | “aaa” |
|  | CreatedDate | DATETIME | Y | N | N |  | TEXT | 12-12-2021 12:12:11 |
|  | UserID | INTEGER | Y | N | N |  | INTEGER | 1 |
| Sqlite scripts | | 1. --Create the table   CREATE TABLE "Sequence" (  "ID" INTEGER NOT NULL UNIQUE,  "SequenceName" TEXT NOT NULL UNIQUE,  "CreatedDate" TEXT NOT NULL,  "UserID" INTEGER NOT NULL,  PRIMARY KEY ("ID" AUTOINCREMENT)  );   1. --Insert a record   INSERT INTO Sequence (SequenceName,CreatedDate, UserID) VALUES ("aaa","2021-11-11 12:12:12 666",1) | | | | | | |
| Tips | |  | | | | | | |
| Notes | | The table operands should include INSERT for a whole record, **DELETE oldest records using quantity,** UPDATE “SequenceName”, “CreateDate” and “UserID” fields following ID, QUERY records using “SequenceName”, “CreatedDate” and quantity. | | | | | | |

## Table SequencePreset

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | SequencePreset | | | | | | |
| Primary Key | | ID | | | | | | |
| Other fields | | SequenceID, Order, Quantity, PresetID | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite type | Example value |
|  | ID | INTEGER | Y | N | Y |  | INTEGER | 1 |
|  | SequenceID | INTEGER | Y | N | N |  | INTEGER | 1 |
|  | Order | INTEGER | Y | N | N |  | INTEGER | 0 |
|  | Quantity | INTEGER | Y | N | N | 1 | INTEGER | 1 |
|  | PresetID | INTEGER | Y | N | N |  | INTEGER | 1 |
| Sqlite Scripts | | 1. --Create the table   CREATE TABLE "SequencePreset" (  "ID" INTEGER NOT NULL UNIQUE,  "SequenceID" INTEGER NOT NULL,  "Order" INTEGER NOT NULL,  "Quantity" INTEGER NOT NULL DEFAULT 1,  "PresetID" INTEGER NOT NULL,  PRIMARY KEY("ID" AUTOINCREMENT),  FOREIGN KEY("SequenceID") REFERENCES "Sequence"("ID") ON DELETE CASCADE,  FOREIGN KEY("PresetID") REFERENCES "WeldRecipe"("ID") ON DELETE CASCADE  ); | | | | | | |
| Tips | |  | | | | | | |
| Notes: | | The table operands should include INSERT for a whole record, **DELETE oldest records using quantity,** UPDATE “Order”, “Quantity” and “PresetID” fields following ID, QUERY records using “SequenceID” and quantity. | | | | | | |

## Table LastOperate

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | LastOperate | | | | | | |
| Primary Key | | -- | | | | | | |
| Other fields | | PartCount, EndCount, OperateType, PresetID, SequenceID | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | PartCount | INTEGER | N | N | Y | 0 | INTEGER | 0 |
|  | EndCount | INTEGER | N | N | Y | 0 | INTEGER | 10 |
|  | OperateType | ENUM | N | N | Y |  | INTEGER | 0 |
|  | PresetID | INTEGER | N | N | Y |  | INTEGER | 1 |
|  | SequenceID | INTEGER | N | N | Y |  | INTEGER | 1 |
| Sqlite Script | | --Create the table  CREATE TABLE "LastOperate" (  "PartCount" INTEGER DEFAULT 0 UNIQUE,  "EndCount" INTEGER DEFAULT 0 UNIQUE,  "OperateType" INTEGER UNIQUE,  "PresetID" INTEGER UNIQUE,  "SequenceID" INTEGER UNIQUE,  FOREIGN KEY("PresetID") REFERENCES "WeldRecipe"("ID") ON DELETE CASCADE); | | | | | | |
| Tips | |  | | | | | | |
| Notes | | The table shall have only one record so the basic operand should have INSERT and DELETE. | | | | | | |

## Table Communication

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | Communication | | | | | | |
| Primary | | -- | | | | | | |
| Other fields | | EthernetType, TCPIPswitch, RemoteDataLogging, GatewayData, GatewayConfig, GatewayCurve, ServerPort, IPAddress | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | EthernetType | ENUM | Y | N | Y | -1 | INTEGER | -1(no data sent) |
|  | TCPIPswitch | BOOLEAN | Y | N | Y | 0 | INTEGER | 0 |
|  | RemoteDataLogging | BOOLEAN | Y | N | Y | 0 | INTEGER | 0 |
|  | GatewayData | BOOLEAN | Y | N | Y | 0 | INTEGER | 0 |
|  | GatewayConfig | BOOLEAN | Y | N | Y | 0 | INTEGER | 0 |
|  | GatewayCurve | BOOLEAN | Y | N | Y | 0 | INTEGER | 0 |
|  | ServerPort | INTEGER | Y | N | Y | 4000 | INTEGER | 4000 |
|  | IPAddress | CHAR(15) | Y | N | Y | 192.168.2.101 | TEXT | 127.0.0.1 |
| Sqlite Scripts | | --Create the table  CREATE TABLE "Communication" (  "EthernetType" INTEGER NOT NULL DEFAULT 1 UNIQUE,  "TCPIPswitch" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "RemoteDataLogging" INTEGER NOT NULL UNIQUE,  "GatewayData" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "GatewayConfig" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "GatewayCurve" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "ServerPort" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "IPAddress" TEXT NOT NULL DEFAULT '127.0.0.1' UNIQUE  ); | | | | | | |
| Tips | |  | | | | | | |
| Notes | | Please ignore it. | | | | | | |

## Table GlobalSetting

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | GlobalSetting | | | | | | |
| Primary Key | | -- | | | | | | |
| Other fields | | HeightEncoder, Seek, FootPedalAbort, LockOnAlarm, GraphData, DoubleHitMode, CoolingType, CoolingDuration, CoolingDelay, PowerSupply, CalibrationAmplitude, ActuatorFrequency, FrequencyOffset, TunePoint | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example value |
|  | HeightEncoder | BOOLEAN | Y | N | Y | 0 | INTEGER | 0(off) |
|  | Seek | BOOLEAN | Y | N | Y | 0 | INTEGER | 0(off) |
|  | FootPedalAbort | BOOLEAN | Y | N | Y | 0 | INTEGER | 0(off) |
|  | LockOnAlarm | BOOLEAN | Y | N | Y | 0 | INTEGER | 0(off) |
|  | GraphData | BOOLEAN | Y | N | Y | 0 | INTEGER | 0(off) |
|  | DoubleHitMode | BOOLEAN | Y | N | Y | 0 | INTEGER | 0(off) |
|  | CoolingType | ENUM | Y | N | Y | 0 | INTEGER | 0 |
|  | CoolingDuration | INTEGER | Y | N | Y | 1.00 | REAL | 1.00(s) |
|  | CoolingDelay | INTEGER | Y | N | Y | 0.00 | REAL | 0.00(s) |
|  | PowerSupply | INTEGER | Y | N | Y | 4000 | INTEGER | 4000(W) |
|  | CalibrationAmplitude | INTEGER | Y | N | Y | 72 | INTEGER | 72(um) |
|  | ActuatorFrequency | INTEGER | Y | N | Y | 2000 | INTEGER | 2000(Hz) |
|  | FrequencyOffset | INTEGER | Y | N | Y | 0 | INTEGER | 0(Hz) |
|  | TunePoint | INTEGER | Y | N | Y | 19950 | INTEGER | 19950(Hz) |
|  | HomePosition | INTEGER | Y | N | Y | 15.00 | REAL | 15.00(mm) |
| Sqlite Scripts | | --Create the table  CREATE TABLE "GlobalSetting" (  "HeightEncoder" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "Seek" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "FootPedalAbort" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "LockOnAlarm" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "GraphData" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "DoubleHitMode" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "CoolingType" INTEGER NOT NULL DEFAULT 0 UNIQUE,  "CoolingDuration" REAL NOT NULL DEFAULT 1.00 UNIQUE,  "CoolingDelay" REAL NOT NULL DEFAULT 0.00 UNIQUE,  "PowerSupply" INTEGER NOT NULL DEFAULT 4000,  "CalibrationAmplitude" INTEGER NOT NULL DEFAULT 72,  "ActuatorFrequency" INTEGER NOT NULL DEFAULT 2000,  "FrequencyOffset" INTEGER NOT NULL DEFAULT 0,  "TunePoint" INTEGER NOT NULL DEFAULT 19950,  "HomePosition" REAL NOT NULL DEFAULT 15.00 UNIQUE  ); | | | | | | |
| Tips | |  | | | | | | |
| Notes | | Please ignore it. | | | | | | |

## Table HeightCalibration

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table Name | | HeightCalibration | | | | | | |
| Primary Key | | ID | | | | | | |
| Other fields | | shim, pressure, HeightValue | | | | | | |
| NO. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | ID | INTEGER | Y | Y | Y |  | INTEGER | 1 |
|  | shim | INTEGER | Y | N | N |  | INTEGER | 1(mm)/6(mm) |
|  | pressure | INTEGER | Y | N | N |  | INTEGER | 10(psi) |
|  | HeightValue | TEXT | Y | N | N |  | TEXT | “1,2,3,4,5,5,4,3,2,1,1,2,3,4,5,5,4,3,2,1” |
| Sqlite Scripts | | --Create the table  CREATE TABLE "HeightCalibration"(  "ID" INTEGER NOT NULL UNIQUE,  "shim" INTEGER NOT NULL,  "pressure" INTEGER NOT NULL,  "HeightValue " TEXT NOT NULL,  PRIMARY KEY("ID" AUTOINCREMENT)  ); | | | | | | |
| Tips | | shim: It’s the shim height. And there are 2 different heights of 1mm and 6mm.  pressure: the maximum pressure is 80psi, and the step is 1, so there are 80 psi in total.  HexHeightValue: the height value under each psi. The value is recorded from 100ms to 2000ms, and the time step is 100ms. So, there have 20 numbers.  There need 80 records in the table. | | | | | | |
| Notes | | Please ignore it | | | | | | |

## Table MaintenanceLog

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | MaintenanceLog | | | | | | |
| Primary Key | | ID | | | | | | |
| Other fields | | MaintenanceType, MaintenanceMsg, CreateDate, UserID | | | | | | |
| No. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | ID | INTEGERT | Y | N | Y | -- | INTEGER | 1 |
|  | MaintenanceType | ENUM | Y | N | N | -- | TEXT | 1(Calibration) |
|  | MaintenanceMsg | TEXT | Y | N | N | -- | TEXT | “Height Calibration process start” |
|  | CreateDate | DATETIME | Y | N | N | -- | TEXT | 12-12-2021 12:12:11 |
|  | UserID | INTEGER | Y | N | N | -- | INTEGER | 1 |
| Sqlite Scripts | | --Create the table  CREATE TABLE "MaintenanceLog" (  "ID" INTEGER NOT NULL UNIQUE,  "MaintenanceType" INTEGER NOT NULL,  "MaintenanceMsg" TEXT NOT NULL,  "CreateDate" TEXT NOT NULL,  "OperatorID" INTEGER NOT NULL,  PRIMARY KEY ("ID" AUTOINCREMENT)  ); | | | | | | |
| Tips | |  | | | | | | |
| Notes: | | Shall be same with Alarm Log | | | | | | |

## Table TeachModeConfiguration

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | TeachModeConfiguration | | | | | | |
| Primary Key | |  | | | | | | |
| Other fields | | TeachModeType, TimePLRG\_STD, TimeMSRG\_STD, PowerPLRG\_STD, PowerMSRG\_STD, PreHeightPLRG\_STD, PreHeightMSRG\_STD, HeightPLRG\_STD, HeightMSRG\_STD, ForcePLRG\_STD, ForceMSRG\_STD, TimePLRG\_AUTO, TimeMSRG\_AUTO, PowerPLRG\_AUTO, PowerMSRG\_AUTO, PreHeightPLRG\_AUTO, PreHeightMSRG\_AUTO, HeightPLRG\_AUTO, HeightMSRG\_AUTO, ForcePLRG\_AUTO, ForceMSRG\_AUTO, TimeConfigPL, TimeConfigMS, PowerConfigPL, PowerConfigMS, PreHeightConfigPL, PreHeightConfigMS, HeightConfigPL, HeightConfigMS, StandardQuantity, AutoQuantity, SigmaQuantity | | | | | | |
| NO. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | TeachModeType | ENUM | Y | N | Y | 0 | INTEGER | 0 |
|  | TimePLRG\_STD | INT | Y | N | Y | 40 | INTEGER | 40 |
|  | TimeMSRG\_STD | INT | Y | N | Y | 40 | INTEGER | 40 |
|  | PowerPLRG\_STD | INT | Y | N | Y | 25 | INTEGER | 25 |
|  | PowerMSRG\_STD | INT | Y | N | Y | 25 | INTEGER | 25 |
|  | PreHeightPLRG\_STD | INT | Y | N | Y | 15 | INTEGER | 15 |
|  | PreHeightMSRG\_STD | INT | Y | N | Y | 15 | INTEGER | 15 |
|  | HeightPLRG\_STD | INT | Y | N | Y | 10 | INTEGER | 10 |
|  | HeightMSRG\_STD | INT | Y | N | Y | 10 | INTEGER | 10 |
|  | StandardQuantity | INT | Y | N | Y | 15 | INTEGER | 15 |
|  | TimePLRG\_AUTO | INT | Y | N | Y | 40 | INTEGER | 40 |
|  | TimeMSRG\_AUTO | INT | Y | N | Y | 40 | INTEGER | 40 |
|  | PowerPLRG\_AUTO | INT | Y | N | Y | 25 | INTEGER | 25 |
|  | PowerMSRG\_AUTO | INT | Y | N | Y | 25 | INTEGER | 25 |
|  | PreHeightPLRG\_AUTO | INT | Y | N | Y | 15 | INTEGER | 15 |
|  | PreHeightMSRG\_AUTO | INT | Y | N | Y | 15 | INTEGER | 15 |
|  | HeightPLRG\_AUTO | INT | Y | N | Y | 10 | INTEGER | 10 |
|  | HeightMSRG\_AUTO | INT | Y | N | Y | 10 | INTEGER | 10 |
|  | AutoQuantity | INT | Y | N | Y | 15 | INTEGER | 15 |
|  | TimeConfigPL | INT | Y | N | Y | 4 | INTEGER | 4 |
|  | TimeConfigMS | INT | Y | N | Y | 4 | INTEGER | 4 |
|  | PowerConfigPL | INT | Y | N | Y | 4 | INTEGER | 4 |
|  | PowerConfigMS | INT | Y | N | Y | 4 | INTEGER | 4 |
|  | PreHeightConfigPL | INT | Y | N | Y | 4 | INTEGER | 4 |
|  | PreHeightConfigMS | INT | Y | N | Y | 4 | INTEGER | 4 |
|  | HeightConfigPL | INT | Y | N | Y | 4 | INTEGER | 4 |
|  | HeightConfigMS | INT | Y | N | Y | 4 | INTEGER | 4 |
|  | SigmaQuantity | INT | Y | N | Y | 15 | INTEGER | 15 |
| Sqlite Scripts | | --Create the table  CREATE TABLE "TeachModeConfiguration" (  "TeachModeType" INTEGER NOT NULL DEFAULT 0,  "TimePLRG\_STD" INTEGER NOT NULL UNIQUE,  "TimeMSRG\_STD" INTEGER NOT NULL DEFAULT 40,  "PowerPLRG\_STD" INTEGER NOT NULL DEFAULT 25,  "PowerMSRG\_STD" INTEGER NOT NULL DEFAULT 25,  "PreHeightPLRG\_STD" INTEGER NOT NULL DEFAULT 15 UNIQUE,  "PreHeightMSRG\_STD" INTEGER NOT NULL DEFAULT 15,  "HeightPLRG\_STD" INTEGER NOT NULL DEFAULT 10,  "HeightMSRG\_STD" INTEGER NOT NULL DEFAULT 10,  "StandardQuantity" INTEGER NOT NULL DEFAULT 15 UNIQUE,  "TimePLRG\_AUTO" INTEGER NOT NULL,  "TimeMSRG\_AUTO" INTEGER NOT NULL DEFAULT 40 UNIQUE,  "PowerPLRG\_AUTO" INTEGER NOT NULL DEFAULT 25 UNIQUE,  "PowerMSRG\_AUTO" INTEGER NOT NULL DEFAULT 25 UNIQUE,  "PreHeightPLRG\_AUTO" INTEGER NOT NULL DEFAULT 15 UNIQUE,  "PreHeightMSRG\_AUTO" INTEGER NOT NULL DEFAULT 15 UNIQUE,  "HeightPLRG\_AUTO" INTEGER NOT NULL DEFAULT 10 UNIQUE,  "HeightMSRG\_AUTO" INTEGER NOT NULL DEFAULT 10 UNIQUE,  "AutoQuantity" INTEGER NOT NULL DEFAULT 15 UNIQUE,  "TimeConfigPL" INTEGER NOT NULL DEFAULT 4 UNIQUE,  "TimeConfigMS" INTEGER NOT NULL DEFAULT 4 UNIQUE,  "PowerConfigPL" INTEGER NOT NULL DEFAULT 4 UNIQUE,  "PowerConfigMS" INTEGER NOT NULL DEFAULT 4 UNIQUE,  "PreHeightConfigPL" INTEGER NOT NULL DEFAULT 4 UNIQUE,  "PreHeightConfigMS" INTEGER NOT NULL DEFAULT 4 UNIQUE,  "HeightConfigPL" INTEGER NOT NULL DEFAULT 4 UNIQUE,  "HeightConfigMS" INTEGER NOT NULL DEFAULT 4 UNIQUE,  "SigmaQuantity" INTEGER NOT NULL DEFAULT 15 UNIQUE  ); | | | | | | |
| Tips | |  | | | | | | |
| Notes: | | Please ignore it | | | | | | |

## Table DBVersion

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table name | | DBVersion | | | | | | |
| Primary Key | | ID | | | | | | |
| Other fields | | VersionNo | | | | | | |
| NO. | Column | Type | Not Null | Auto Increment | Unique | Default | Sqlite Type | Example Value |
|  | VersionNo | TEXT | Y | N | Y |  | TEXT | 1.0 |
| Sqlite Scripts | | --Create the table  CREATE TABLE "DBVersion" (  "VersionNo" TEXT NOT NULL UNIQUE  ); | | | | | | |
| Tips | |  | | | | | | |
| Notes | | It should be a database key attribute for the whole system. We will manage database using it. | | | | | | |

# DataTask

## General requirements

1. When class DataTask is created, it should open database “sample\_l20\_base.db” using DBAccessL20DB::ConnectDB
2. When class DataTask is created, it should get message queue id of "/Control"
3. When class DataTask is created, it should get message queue id of "/Data"
4. When class DataTask is created, it should get message queue id of "/Request"
5. When class DataTask is destroyed, it should close database “sample\_l20\_base.db” using DBAccessL20DB::CloseDataBaseConnection

## Message Processing

The DataTask should employ 3 separate message queues based on priority to allow other tasks indirect access to the data storage for reading as well as writing.

1. DataTask should process all messages from the CONTROL queue first.
2. Then DataTask should process messages from the DATA queue, then it should check for new messages from the CONTROL queue.
3. Then DataTask should process messages from the REQUEST queue, then it should check for new messages from the CONTROL queue and the DATA queue.



1. The message processing flow of the data task is triggered by event.
2. The message struct should be defined in Common.h as

struct MESSAGE

{

UINT32 msgID;

char Buffer[MAX\_SIZE\_OF\_MSG\_LENGTH - sizeof(msgID)];

};

1. The length of structure should not be out of the Buffer range.

## Database Processing

1. DataTask should provide method to open the database.
2. DataTask should provide method to close the database.
3. DataTask should provide method to insert new record.
4. Data for WeldRecipe should be extracted from message queue using struct type WeldRecipeSC

Comments from Jerry: With meeting discussion with WindRiver and Liu Qun, there are three solutions for the data exchange between control task and data task as following…

* + - 1. Keep global variables CommonProperty::WeldResult and CommonProperty::WeldSignatureVector as the data source for the data exchange between control task and data task. If we still stick to send data using message queue, the Weld Signature size will be out of message queue (1024).
      2. In order to make sure the data integrity without any missing, there is the serial process between control task and data task using message queue. At the begin the control task send the request command to data task and wait for the response command from data task until the data is inserted into database.
      3. It seems the size of message queue is not able to save the weld signature data. In order to keep the data process with high performance, there should be a round-robin queue between control task and data task so control task and data task can run each other parallel. All the data shall be saved in the round-robin queue temporarily.

In order to keep Branson be better policy, let’s keep 1# to implement weld result and weld signature saving with data task.

1. Data for WeldResult should be extracted from CommonProperty::WeldResult
2. Data for WeldResultSignature should be extracted from CommonProperty::WeldSignatureVector
3. The size of vector in CommonProperty::WeldSignatureVector should not exceed 200.
4. When the number of records exceeds the limit of table “WeldResult” and “WeldResultSignature”, the oldest record should be removed.
5. ID of table “WeldResult” and “WeldResultSignature” should be with long long type

Comments from Jerry: We don’t need to restart incremental ID from 1 when it reaches out the table limit. With the table analysis, there is only weld result ID will be related in Alarm table and Weld Signature table. The code needs to handle with weld result ID using long long data type while the data processing for Alarm table and Weld Signature table.

1. When the number of records reaches the limit of table “WeldRecipe”, no more record should be insert.
2. DataTask should provide method to update the record for table Recipe, meanwhile, the column DataTime should be update following current time stamp.
3. The record to be updated for table Recipe is selected by ID using value from m\_RecipeNumber of WeldRecipeSC.
4. DataTask should provide method to query 50 records using ID from table WeldResult.
5. The result of querying from table WeldResult should be parsed into CommonProperty::WeldResultForUI[50].
6. DataTask should provide method to query record of WeldGraph from table WeldResultSignature.
7. The result of querying from table WeldResultSignature should be...
8. DataTask should provide method to query for a record using ID of WeldRecipeSC from table WeldRecipe.
9. The result of querying from table WeldRecipe should be …
10. DataTask should provide method to query for all the 1000 records from table WeldRecipe.