**RoboPlus Manager 2.0**

**인수 인계**

|  |  |
| --- | --- |
| 작성자 | 오재홍 |
| 최초 작성 | 2015년 12월 7일 |
| 최근 수정 | 2015년 12월 15일 |

# 개발 소스에 관한 기본 설명

소스 위치 (<https://github.com/ojh6t3k/RoboPlusManager>)

위 URL 접속 후 Download ZIP버튼을 통해 최근까지 개발한 소스를 다운로드 받을 수 있다.

폴더 구조 설명

* AndroidPlugin : CommSocket용 Android Plugin 개발 소스 (Eclipse기반)
* OSXPlugin : CommSocket용 OSX Plugin 개발 소스 (Xcode 기반)
* Assets : Unity3D용 개발 소스
  + Assets/RoboPlusManager: RoboPlusManager개발과 관련된 소스
* ProjectSettings: Unity3D용 설정 정보

목차

[개발 소스에 관한 기본 설명 2](#_Toc437964205)

[Class dependency 5](#_Toc437964206)

[CommSocket 6](#_Toc437964207)

[CommDevice 8](#_Toc437964208)

[CommProtocol 9](#_Toc437964209)

[Result 11](#_Toc437964210)

[CommProduct 12](#_Toc437964211)

[ProductInfo 14](#_Toc437964212)

[ControlUIInfo 16](#_Toc437964213)

[ControlItemInfo 17](#_Toc437964214)

[Product XML 18](#_Toc437964215)

[ProductManager 21](#_Toc437964216)

[LocalizationManager 23](#_Toc437964217)

[LocalizedText 27](#_Toc437964218)

[LanguageInfo 28](#_Toc437964219)

[UpdownValue 29](#_Toc437964220)

[RepeatButton 31](#_Toc437964221)

[DialControl 32](#_Toc437964222)

[ListView 34](#_Toc437964223)

[ListItem 36](#_Toc437964224)

[ControlUI 38](#_Toc437964225)

[ControlUIManager 40](#_Toc437964226)

[DefaultUI 41](#_Toc437964227)

[CommunicationUI 42](#_Toc437964228)

[AlramUI 44](#_Toc437964229)

[AuxDeviceUI 46](#_Toc437964230)

[DriveUI 48](#_Toc437964231)

[OperateUI 51](#_Toc437964232)

[TorqueUI 52](#_Toc437964233)

[RoboPlusManager 54](#_Toc437964234)

[CommProductUI 56](#_Toc437964235)

# Class dependency

**CommSocket**

**CommDevice**

**CommProtocol**

**Result**

**CommProduct**

**ProductInfo**

**ControlUIInfo**

**ControlItemInfo**

**ProductManager**

**LocalizationManager**

**LocalizedText**

**LanguageInfo**

**UpdownValue**

**RepeatButton**

**DialControl**

**ListView**

**ListItem**

**ControlUI**

**ControlUIManager**

**DefaultUI**

**CommunicationUI**

**AlramUI**

**AuxDeviceUI**

**DriveUI**

**OperateUI**

**TorqueUI**

**RoboPlusManager**

**CommProductUI**

**UI 관련 Class**

**Core 관련 Class**

# CommSocket

통신 장치 사용/해제 관리용 (CommSocket.cs)

의존성

* CommDevice
* AndroidPlugin / OSX Plugin

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| foundDevices | Search를 통해 찾은 장치 정보를 저장하는 리스트  장치 정보 포맷은 CommDevice에서 확인할 수 있다. |
| device | Open할 통신 장치 정보  Inspector에서는 foundDevices중에서 선택한 정보를 입력한다. |
| serialBaudrate | Serial 장치의 경우 사용되는 Baudrate 정보 |
| searchTimeout | Search시 사용할 시간 (단위: sec)  Search 시작 후 CancelSearch를 별도로 하지 않는다면 이 시간이 지난 후 자동으로 Search가 종료되고 OnSearchCompleted 이벤트가 발생한다. |
| isOpen | 통신 연결 유무를 알려줌 |
| isSupportBLE | 현재 플랫폼이 BLE를 지원하는지를 알려줌 |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| SetSerialBaudrate | serialBaudrate를 설정 |
| Open | device 정보를 이용해서 장치 사용을 시도한다.  이 시도의 결과는 다음의 이벤트로 알 수 있다.  OnOpen: 통신 장치 사용 성공  OnOpenFailed: 통신 장치 사용 실패 |
| Close | 연결된 통신 장치 사용을 종료한다. |
| StartSearch | 사용 가능한 통신 장치를 찾는다.  CommDevice Type을 이용해서 필터링할 수 있다.  만약, 장치가 검색된 경우 OnFoundDevice 이벤트가 발생함. |
| CancelSearch | 통신 장치 검색을 강제로 종료한다.  현재 Search중이라면 OnSearchCompleted 이벤트가 발생한다. |
| Write | Byte[] 타입 데이터를 전송한다.  만약, 실패한다면 OnErrorClosed 이벤트가 발생하고 자동으로 통신 장치는 Close된다. |
| Read | Byte[] 타입 데이터를 반환한다.  만약, 데이터가 없다면 null을 반환한다.  만약, 실패한다면 OnErrorClosed 이벤트가 발생하고 자동으로 통신 장치는 Close된다. |

Event

|  |  |
| --- | --- |
| 이름 | 설명 |
| OnOpen | Open으로 통신 장치 사용 시도 시 성공하면 발생한다. |
| OnClose | Close로 통신 장치 사용 종료 시 발생한다. |
| OnOpenFailed | Open으로 통신 장치 사용 시도 시 실패하면 발생한다. |
| OnErrorClosed | 예외 상황으로 통신 장치 사용이 종료되면 발생한다. |
| OnFoundDevice | Search 후 foundDevice의 리스트가 변경되면 발생한다. |
| OnSearchCompleted | Search가 종료되면 발생한다. |

Details

* 플랫폼 별 사용 가능 장치 타입
  + Windows: Serial (BT는 Serial로 인식됨)
  + Mac: Serial (BT는 Serial로 인식됨), BLE
  + Android: BT, BLE
  + iOS: 아직 구현 안됨

# CommDevice

통신 장치 정보 관리용 (CommSocket.cs)

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| name | 장치 이름 |
| displayName | type과 name을 합쳐서 반환 |
| type | 통신 장치 타입  Serial: 시리얼 포트  BT: Bluetooth 2.0  BLE: Bluetooth Low Energy |
| address | 통신 장치 연결 시 사용되는 정보 |
| args | 장치 타입마다 다른 부가 정보  Serial: Baudrate  BT: 사용 안함  BLE: UUID |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| CommDeivce | 생성자  다른 CommDevice를 인자로 사용 시 복사된다. |
| Equals | 다른 CommDevice와 동일한 지 비교한다. |

# CommProtocol

통신 프로토콜 Encoding/Decoding (CommProtocol.cs)

의존성

* CommSocket
* Result

Define

|  |  |
| --- | --- |
| 이름 | 설명 |
| BROADCAST\_ID | Broadcast ID (254) |
| CM\_ID | 제어기 ID (200) |
| MAX\_ID | 사용 가능한 최대 ID (252) |

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| protocol | 사용할 프로토콜  CM: 제어기 1.0 프로토콜  DXL: 다이나믹셀 1.0 프로토콜  DXL2: 다이나믹셀 2.0 프로토콜  UnKnown: 정의 안됨 |
| socket | 통신에 사용할 CommSocket |
| debug | 설정 시 통신에 사용되는 데이터 패킷을 Console창을 통해 확인할 수 있다. |
| instance | 전역 참조 시 사용됨 |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| SetProtocolDXL | Protocol을 DXL로 설정 |
| SetProtocolDXL2 | Protocol을 DXL2로 설정 |
| Ping | 해당 id를 가진 장치가 있는지 확인한다.  결과는 OnResult 이벤트로 확인 |
| AskWho | 해당 id를 가진 장치의 모델 번호와 버전을 확인한다.  결과는 OnResult 이벤트로 확인 |
| Read | 해당 id를 가진 장치로부터 데이터 읽기 시도  결과는 OnResult 이벤트로 확인 |
| Write | 해당 id를 가진 장치에 데이터 쓰기 시도  결과는 OnResult 이벤트로 확인 |
| Word2Byte | 16bit 데이터를 2개의 8bit 데이터로 분리 |
| Byte2Word | 2개의 8bit 데이터를 16bit 데이터로 병합 |

Event

|  |  |
| --- | --- |
| 이름 | 설명 |
| OnResult | 장치와 통신 시도 시 결과 이벤트  정보는 Result 타입으로 전달됨 |

Details

* 참고 사항
  + CM 프로토콜은 구현 안됨

# Result

통신 프로토콜 결과 (CommProtocol.cs)

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| query | 장치 전송 시 사용된 정보  None: 없음  Ping: Ping 결과  AskWho: AskWho 결과  Read: Read 결과  Write: Write 결과 |
| id | 본 결과를 준 장치 id |
| txFail | 프로토콜 전송 실패 여부 |
| timeout | 응답이 없어서 프로토콜 수신 실패 |
| rxFail | 응답은 있었으나 프로토콜 해석 실패 |
| statusError | 수신 프로토콜에서 상태 정보 |
| address | 받은 데이터의 시작 address |
| parameters | 받은 데이터 |

# CommProduct

Product 정보 기반으로 통신 (CommProduct.cs)

의존성

* CommProtocol
* Result
* ProductInfo
* ControlItemInfo

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| connected | 장치와 연결 여부 확인 |
| id | 통신에 사용 중인 id |
| version | 장치로부터 확인된 펌웨어 버전 |
| model | 장치로부터 확인된 모델 번호 |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| Connect | 설정된 id로 연결 시도  OnConnected 이벤트 발생 시 연결 성공  OnConnectionFailed 이벤트 발생 시 연결 실패 |
| Disconnect | 연결 해제  물리적 통신 연결을 해제하는 것이 아닌 주기적 통신만 종료  OnDisconnected 이벤트 발생 |
| Run | 주기적 통신 시작 |
| Stop | 주기적 통신 멈춤 |
| ClearItem | 주기적 통신을 위해 등록한 Read/Write 컨트롤 테이블 모두 제거 |
| SetWriteItem | 주기적 통신을 위해 등록한 기존 Write 컨트롤 테이블을 모두 제거하고 새로 등록  Write는 반복되지 않고 수행한 후 자동으로 제거된다. |
| AddWriteItem | 주기적 통신을 위해 Write 컨트롤 테이블을 기존 목록에 추가 등록  Write는 반복되지 않고 수행한 후 자동으로 제거된다. |
| AddReadItem | 주기적 통신을 위해 Read 컨트롤 테이블을 기존 목록에 추가  Read는 계속 반복된다. |
| RemoveReadItem | 주기적 통신을 위해 등록된 Read 컨트롤 테이블을 목록에서 제거 |

Event

|  |  |
| --- | --- |
| 이름 | 설명 |
| OnConnected | Connect시도 후 성공하면 발생  해당 CommProduct를 인자로 전달 |
| OnConnectionFailed | Connect시도 후 실패하면 발생  해당 CommProduct를 인자로 전달 |
| OnDisconnected | Disconnect시도 후 발생  해당 CommProduct를 인자로 전달 |
| OnLostConnection | 예외 상황으로 Disconnect되었을 때 발생  해당 CommProduct를 인자로 전달 |
| OnReadUpdate | 주기적 통신에서 Read 컨트롤 테이블 수행 완료 시 발생  Read는 등록된 목록을 한꺼번에 수행한다. |
| OnWriteCompleted | 주기적 통신에서 Write 컨트롤 테이블 수행 완료 시 발생  Wirte는 등록된 목록을 하나씩 수행한다. |

# ProductInfo

Product 정보 (ProductManager.cs)

의존성

* ControlUIInfo
* ControlItemInfo
* Product XML

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| name | 장치의 이름 |
| key | DB에서 구별되는 고유 식별자 |
| type | 장치의 종류  CM: 제어기  DXL: 다이나믹셀 |
| model | XML상의 모델 번호 |
| sprite | 제품 이미지 |
| firmware | 펌웨어 복구/업데이트에 사용할 binary 파일 |
| firmwareVersion | 펌웨어 파일의 버전 |
| firmwareAddress | 펌웨어 복구/업데이트에 사용할 부트로더 어드레스 |
| calibration | 켈리브레이션 시 사용할 binary 파일 |
| calibrationVersion | 캘리브레이션 파일의 버전 |
| protocol | 통신 시 사용되는 프로토콜  CM: 제어기 1.0 프로토콜  DXL: 다이나믹셀 1.0 프로토콜  DXL2: 다이나믹셀 2.0 프로토콜 |
| uiList | 통신 UI에 사용할 리스트  상세 내용은 ControlUIInfo 참조 |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| GetControlUIInfo | uiClass이름으로 uiList에서 ControlUIInfo를 얻어옴 |

# ControlUIInfo

Product 통신에 사용될 UI정보 (ProductManager.cs)

의존성

* ControlItemInfo
* Product XML

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| name | UI의 이름 |
| icon | 아이콘 |
| uiClass | UI구성에 사용될 실제 Class 이름 |
| version | UI 버전 |
| uiParameters | UI 구성에 참고할 정보 |
| uiItems | ControlItemInfo 리스트 |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| GetUIItem | 이름으로 uiItems에서 ControlItemInfo를 얻어옴 |

# ControlItemInfo

Product 통신에 사용될 아이템 정보 (ProductManager.cs)

의존성

* Product XML

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| name | 아이템의 이름 |
| address | 컨트롤 테이블 어드레스 |
| access | 접근 한정자  R: Read 프로토콜만 사용 가능  RW: Read/Write 프로토콜 모두 사용 가능  W: Write 프로토콜만 사용 가능 |
| savable | EEPROM Save 가능 여부 |
| bytes | 데이터의 바이트 크기 |
| defaultValue | Reset시 설정될 초기 값 |
| minValue | 최소 값 |
| maxValue | 최대 값 |
| writeValue | Write 프로토콜로 시도되는 값 |
| value | Read 프로토콜로 설정되는 값 |
| update | Read 프로토콜로 갱신 여부 |
| modify | Write 프로토콜 시도 여부 |

# Product XML

Product 표현에 필요한 정보를 XML로 표현하는 규격

의존성

* ProductManager

<Product>

Attribute

|  |  |
| --- | --- |
| 이름 | 설명 |
| key | DB 고유 식별자  용도  -펌웨어 파일 이름  -이미지 이름 |
| name | 장치의 이름 |
| type | 장치의 종류  CM: 제어기  DXL: 다이나믹셀 |
| model | 모델 번호 |

예) <Product key="AX12A" name="AX-12A" type="DXL" model="12">

하위 Schema

* <Firmware>
* <Control>

<Firmware>

Attribute

|  |  |
| --- | --- |
| 이름 | 설명 |
| version | 펌웨어 파일의 버전 |

예) <Firmware version="28"></Firmware>

하위 Schema

* 없음

<Control>

Attribute

|  |  |
| --- | --- |
| 이름 | 설명 |
| protocol | 통신시 사용되는 프로토콜  CM: 제어기 1.0 프로토콜  DXL: 다이나믹셀 1.0 프로토콜  DXL2: 다이나믹셀 2.0 프로토콜 |

예) <Control protocol="DXL">

하위 Schema

* <UI>

<UI>

Attribute

|  |  |
| --- | --- |
| 이름 | 설명 |
| icon | 아이콘으로 사용될 이미지 이름 |
| name | UI 이름 |
| class | 실제로 작동할 Class 이름 |
| version | 용도별로 구분할 UI 버전 |
| param | UI 구성에 사용할 파라미터  파라미터 포맷은 class별로 다르다. |

예) <UI icon="TestIcon\_3" name="Drive" class="DriveUI" version="1" param="-150~150,0~114">

하위 Schema

* <Item>

<Item>

Attribute

|  |  |
| --- | --- |
| 이름 | 설명 |
| name | 이름 |
| address | 컨트롤 테이블 주소 |
| access | 접근 한정자  r: Read only  rw: Read/Write  w: Write only |
| save | EEPROM 저장 여부 true/false |
| byte | 데이터의 byte 수 |
| default | Reset시 사용할 값  사용 안 할 수도 있음 |
| range | 값의 범위  ‘~’ 구분 |
| continue | XML로드 시 현재 정보를 바탕으로 반복여부 설정  반복 시 address를 byte값 단위로 증가시킨다.  ~ 구분: 시작과 끝  Start address = address + 시작 \* byte  ~가 없는 경우는 반복 회수로 사용 0에서 시작하여 값 - 1까지 반복 |

예) <Item name="CWAngleLimit" address="6" access="rw" save="true" byte="2" default="0" range="0~1023"></Item>

하위 Schema

* 없음

# ProductManager

Product XML을 데이터로 표현 (ProductManager.cs)

의존성

* ProductInfo
* ControlUIInfo
* ControlItemInfo
* Product XML

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| productList | Product XML로 로드된 ProductInfo 리스트 |
| manager | 전역 참조를 위한 인스턴스 반환 |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| GetProductInfo | model번호로 productList에서 ProductInfo 정보 반환  없으면 null 반환 |

Details

* 참고 사항
  + Scene시작 시 XML Load됨
* XML파일 위치 규칙
  + Resources/Product 폴더 아래에 있어야 함
  + Resources/Product/Calibration: Calibration 파일 위치
  + Resources/Product/Firmware: 펌웨어 파일 위치
  + Resources/Product/Icon: 아이콘 파일 위치
  + Resources/Product/Image: 이미지 파일 위치

# LocalizationManager

다국어 테이블 관리 (LocalizationManager.cs)

의존성

* LanguageInfo
* LocalizedText

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| table | 다국어 테이블 파일 |
| newLineText | 테이블 내용에서 NewLine으로 인식할 텍스트 |
| languages | 테이블에서 로드할 언어 정보 |
| manager | 전역 참조할 인스턴스 반환자 |
| currentLanguage | LocalizedText에 영향을 줄 Language 선택  ISO 3 letters방식으로 인자를 사용한다. |
| font | 현재 설정할 언어 정보의 폰트 반환 |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| ApplyLanguage | 현재 설정된 Language로 수동으로 LocalizedText에 영향을 줌 |
| GetLocalizedText | 텍스트 key를 바탕으로 현재 설정된 Language Value를 얻어옴 |

Details

* 참고 사항
  + 다국어 테이블은 Scene시작 시에만 로드된다.
  + Lauguage 리스트는 다국어 테이블로부터 얻는 것이 아니고 미리 리스트를 만들어 놓으면 다국어 테이블로부터 필요한 내용이 채워지는 방식이다.
* 다국어 테이블 포맷
  + 각 행은 NewLine으로 구분된다.
  + 각 행의 항목은 tab으로 구분된다.
  + 각 행 별로 항목의 수는 같아야 한다.
  + 제일 첫 행은 ISO 3 letters기반의 언어 이름이 들어간다.
  + 각 행의 첫 항목은 key가 된다.
  + 언어 별 key값은 특수 문자가 들어간다.
    - NewLineText: 내용을 NewLine으로 구분하기 위함 기본값 {/n}
    - Format Text: C# String Format과 같은 규격
* Google 스프레드시트를 이용한 변환 Script

|  |
| --- |
| function onOpen()  {  var ss = SpreadsheetApp.getActiveSpreadsheet();  var myMenuEntries = [{name: "Make Localization Text", functionName: "saveAsTSV"}];  ss.addMenu("My Tool", myMenuEntries);  };  function saveAsTSV()  {  var ss = SpreadsheetApp.getActiveSpreadsheet();  var sheets = ss.getSheets();  for (var i = 0 ; i < sheets.length ; i++)  {  var sheet = sheets[i];  fileName = "Localizations.txt";  var tsvFile = convertRangeToTsvFile\_(fileName, sheet);  DriveApp.createFile(fileName, tsvFile);  }    Browser.msgBox('Success to make Localizations.txt');  }  function convertRangeToTsvFile\_(csvFileName, sheet)  {  // get available data range in the spreadsheet  var activeRange = sheet.getDataRange();  try  {  var data = activeRange.getValues();  var tsvFile = undefined;  // loop through the data in the range and build a string with the tsv data  if (data.length > 1)  {  var tsv = "";  for (var row = 0; row < data.length; row++)  {  for (var col = 0; col < data[row].length; col++)  {  data[row][col] = data[row][col].toString().replace("\n", "{/n}");  }  // join each row's columns  tsv += data[row].join("\t");    // add a carriage return to end of each row, except for the last one  if (row < data.length-1)  {  tsv += "\r\n";  }  }  tsvFile = tsv;  }  return tsvFile;  }  catch(err)  {  Logger.log(err);  Browser.msgBox(err);  }  } |

# LocalizedText

UGUI의 Text 내용을 LocalizationManager로부터 얻어온다. (LocalizedText.cs)

의존성

* UGUI Text
* LocalizationManager

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| key | 테이블 key |
| uiText | 적용할 UGUI Text |
| text | LocalizationManager에 설정된 언어 정보를 기반으로 key에 해당되는 텍스트 반환 |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| ApplyLanguage | 현재 설정된 Language로 수동으로 uiText에 영향을 줌 |

Details

* 참고 사항
  + String Format을 적용할 시에는 수동으로 LocalizedText로부터 key에 해당되는 텍스트를 받아 Format을 적용한 뒤 UGUI Text에 넣어야 한다. 이를 위해서는 uiText를 null로 만들어 자동으로 적용되는 것을 막는다.

# LanguageInfo

LocalizationManager에서 언어 구분 시 사용되는 정보 (LocalizationManager.cs)

의존성

* 없음

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| name | 언어 이름 |
| Iso3Code | ISO 3 letters기반 언어 코드 |
| font | 언어별로 사용할 폰트 |
| image | 언어별 이미지 |
| table | 다국어 테이블로부터 로드된 key값기반 테이블  HashTable이기에 각 항목은 key값으로 찾는다. |
| index | 다국어 테이블 상으로 기록된 항목 순서 |

# UpdownValue

Up/Down 버튼에 의해 값이 증감되는 방식의 UI (UpdownValue.cs)

의존성

* UGUI InputField

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| interactable | 입력 활성/비활성 |
| inputField | UGUI InputField 연결 |
| initValue | 초기 값 |
| unitValue | 증가/감소 단위 값 |
| minValue | 최소 값 |
| maxValue | 최대 값 |
| format | 값 표현 방식 C# ToString Format |
| Value | 현재 값 |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| OnIncrease | 현재 값을 unitValue만큼 증가 시킴 |
| OnDecrease | 현재 값을 unitValue만큼 감소 시킴 |

Event

|  |  |
| --- | --- |
| 이름 | 설명 |
| OnChangedValue | 값 변경 시 이벤트 발생 |

Details

* Up/down Button 연결
  + Button의 OnClick 이벤트에 OnIncrease, OnDecrease를 연결시켜 사용한다.

# RepeatButton

누르고 있는 동안을 체크할 수 있는 Button (RepeatButton.cs)

의존성

* UGUI

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| interactable | 반응 활성/비활성 |
| delayTime | 누르고 있는 동안 이벤트 발생할 주기 (단위 sec) |

Event

|  |  |
| --- | --- |
| 이름 | 설명 |
| OnRepeated | 누르고 있는 동안 발생되는 이벤트 |

Details

* 참고 사항
  + Button이 아닌 일반 Image에도 연결하여 사용할 수 있다.

# DialControl

Dial방식으로 돌리는 UI (DialControl.cs)

의존성

* UGUI

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| interactable | 입력 활성/비활성 |
| knob | 실제로 돌려지는 Image  DialControl과 다른 위치에 있어야 한다. |
| centerAngle | 각도 중간 값 |
| minAngle | 각도 최소 값  이 값 밑으로 회전 안 함 |
| maxAngle | 각도 최대 값  이 값 위로 회전 안 함 |
| cw | 회전에 따른 Angle값의 증감 방향 바꿈 |
| centerValue | 각도와 맵핑할 중간 값 |
| minValue | 각도와 맵핑할 최소 값 |
| maxValue | 각도와 맵핑할 최대 값 |
| angle | 현재 knob회전에 대한 angle값  값을 수정하면 knob이 회전한다. |
| Value | 현재 knob회전에 대한 Value값  값을 수정하면 knob이 회전한다. |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| Reset | 현재 설정된 값 기준으로 리셋 |

Event

|  |  |
| --- | --- |
| 이름 | 설명 |
| OnChangedValue | Knob 회전 시 이벤트 발생 |

# ListView

List방식의 UI (ListView.cs)

의존성

* UGUI
* ListItem

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| itemPanel | 리스트를 표현할 View |
| itemCount | ListItem의 개수 |
| createdItem | 마지막으로 추가된 Item |
| selectedItem | 선택된 item  ListItem으로 역으로 선택 가능 |
| selectedIndex | 선택된 item의 index  Index로 역으로 선택 가능 |
| items | 추가된 item을 ListItem 배열로 반환 |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| ClearItem | 추가된 모든 item 삭제 |
| AddItem | Item 추가 |
| InsertItem | 현재 선택된 item 앞에 삽입 |
| RemoveItem | 현재 선택된 item 삭제  ListItem을 인자로 하면 특정 item 삭제 가능 |

Event

|  |  |
| --- | --- |
| 이름 | 설명 |
| OnChangedSelection | Item 선택 변경 시 발생 |

Details

* List 형으로 만들기 위한 UGUI 필요 요소
  + ScrollRect: List view를 scroll시키기 위해 필요
  + Mask: Content view를 Scroll view만큼만 보여주기 위해 필요
  + Vertical Layout Group: 하위 요소를 Vertical Grid형태로 정렬시키기 위해 필요
  + Content Size Fitter: Content view를 하위 요소들의 크기 만큼 조정하기 위해 필요
    - 하위 요소들은 반드시 Layout Element를 가져서 크기를 지정해야 한다.
* 단일 Item 선택만 가능

# ListItem

ListView에서 사용하는 Item 형식 (ListItem.cs)

의존성

* UGUI
* ListView

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| owner | Item을 소유한 ListView  ListView에 의해 자동 할당됨 |
| imgae | UGUI Image  사용하지 않을 수 있다. |
| textList | UGUI Text 리스트  최소 1개는 있어야 한다. |
| data | 사용자 데이터 연결 가능 |
| index | ListView에 속한 상태에서의 순서 |
| selected | 선택 여부  설정을 통해 강제 선택 가능 |

Details

* ListItem을 만들기 위한 UGUI 필요 요소
  + 반드시 UGUI Toggle로 만들어야 한다.
  + Layout Element를 추가시켜서 크기를 지정해야 한다.
  + 필요에 따라 Image나 Text를 추가한다.
    - 열 정렬은 Anchor를 이용해서 직접 설정해야 한다.
* 사용 시 Intantiate를 이용해서 복제하여 ListView에 추가해야 한다.
  + Prefab으로 만들어 놓고 사용하거나 Scene에 넣은 후 숨겨서 사용할 수 있다.

# ControlUI

ControlUIInfo를 GUI와 연결하기 위한 Base class. UI로 CommProduct를 제어하기 위해서는 이 class를 상속받아 구현해야 한다.

의존성

* ControlUIInfo
* CommProduct

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| uiClass | ControlUIInfo에서 class 구별자로 사용되는 uiClass 이름 (string type) |
| active | 현재 component가 소속된 gameobject의 active 설정  Hide/Show개념으로 사용 가능 |
| commProduct | ControlUIInfo를 연결할 CommProduct 설정  Null을 사용하면 연결 해제됨. |
| uiInfo | 현재 설정된 ControlUIInfo 설정/확인  Null이면 설정 해제됨 |

Virtual Method (상속한 Class에서 구현해야 할)

|  |  |
| --- | --- |
| 이름 | 설명 |
| OnInitialize | Awake시 호출됨.  시작 시 초기화 루틴 구현 |
| OnSetCommProduct | commProduct로 CommProduct가 설정되었을 때 호출됨  XML로부터 로드된 ControlUIInfo와 설정된 CommProduct를 연결할 루틴 구현 |
| OnSetUIInfo | uiInfo로 ControlUIInfo가 설정되었을 때 호출됨  GUI요소를 초기화시킬 루틴 구현 |
| OnUpdateUI | ControlUIInfo가 CommProduct에 의해 ReadUpdate되었을 때 호출됨  변경된 값을 GUI에 반영할 루틴 구현 |
| OnWriteDone | ControlUIIno가 CommProduct에 의해 WriteCompleted가 되었을 때 호출됨  사용자 입력이 CommProduct에 전달된 후 실행할 루틴 구현 |

Protected Method (상속한 Class만 사용 가능)

[정의]

-Value: 프로토콜에서 사용하는 XML에 정의된 값

-SignValue: Min/Max에 의해 CenterValue를 0으로 Mapping된 Value

-Unit: GUI를 표현하기 위해 물리적 단위로 환산한 값

|  |  |
| --- | --- |
| 이름 | 설명 |
| CenterValue | ControlItem의 Min Value와 Max Value의 중간 값을 반환 |
| MaxSignValue | SignValue란 CenterValue를 0으로 했을 때 Min과 Max의 범위 값이다.  MaxSignValue란 CenterValue를 0으로 했을 때의 Max값을 말한다. |
| Value2Unit | UI별로 설정된 Unit(물리적 단위)로 Value를 변경한 것을 말한다. |
| Unit2Value | Unit값을 Value값으로 변경 |
| Value2SignValue | Value를 SignValue로 변경 |
| SignValue2Value | SignValue를 Value로 변경 |
| Value2SignUnit | Value를 SignValue로 바꾸고 그것을 다시 Unit으로 변경함 |

Details

* GUI와 CommProduct를 연결하기 위해서는 반드시 ControlUI를 상속받아 구현한다.

# ControlUIManager

CommProduct와 구현된 ControlUI를 연결시켜주는 대리자

의존성

* CommProduct
* ControlUI(상속받은 Class 포함)

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| selectedUI | ControlUIInfo의 설정/반환 |

Details

* ControlUIManager의 역할은 다음과 같다.
  + ControlUIInfo를 ControlUIManager에 알려주면 등록된 ControlUI 중 ControlUIInfo와 일치하는 ControlUI를 찾아 CommProduct와 연결시켜준다.
  + 만약, ControlUIManager에 등록된 ControlUI중 설정한 ControlUIIno가 없다면 DefaultUI에 연결한다.

# DefaultUI

ControlUI 상속 Class로써 ControlUIIno의 내용을 GUI Text로 보여주는 역할

의존성

* ControlUI

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| uiText | UGUI Text를 연결하면 현재 선택된 ControlUIInfo의 내용을 Text로 볼 수 있음. |

Details

* ControlUIManager의 DefaultUI로 사용하면 디버그용으로 사용 가능하다.

# CommunicationUI

ControlUIInfo 중 Communication 관련 내용에 대응하기 위한 Class

의존성

* ControlUI
* UpdownValue
* UGUI

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| uiSave | GUI를 통해 설정된 값을 EEPROM에 Save하기 위한 버튼 |
| uiBaudrate | ControlItem 중 Baudrate항목을 설정하기 위한 GUI |
| uiBaudrate2 | UI Version 2 타입에 대응하기 위한 Baudrate항목 설정용 GUI |
| uiID | ControlItem 중 ID항목을 설정하기 위한 GUI |
| uiReturnDelayTime | ControlItem 중 Return Delay 항목을 설정하기 위한 GUI |
| uiReturnLevel0 | ControlItem 중 Return Level을 0으로 설정하기 위한 GUI |
| uiReturnLevel1 | ControlItem 중 Return Level을 1로 설정하기 위한 GUI |
| uiReturnLevel2 | ControlItem 중 Return Level을 2로 설정하기 위한 GUI |
| uiBPS | 설정된 uiBaudrate를 bps단위로 표현하기 위한 GUI |
| uiDelayTime | uiReturnDelayTime을 time단위로 표현하기 위한 GUI |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| OnReset | Communication에 속하는 모든 ControlItem항목을 default Value로 바꾼다.  Reset버튼과 연결하여 사용할 수 있음 |
| OnSave | 변경된 값을 CommProduct에 전송하여 EEPROM에 Save한다.  Save버튼과 연결하여 사용할 수 있음 |
| OnChangedBaudrate | uiBaudrate의 변경된 값을 연결된 ControlItem에 반영한다.  uiBaudrate Event와 연결하여 사용할 수 있음 |
| OnChangedBaudrate2 | uiBaudrate2의 변경된 값을 연결된 ControlItem에 반영한다.  uiBaudrate2 Event와 연결하여 사용할 수 있음  만약, 장치가 검색된 경우 OnFoundDevice 이벤트가 발생함. |
| OnChangedID | uiID의 변경된 값을 연결된 ControlItem에 반영한다.  uiID Event와 연결하여 사용할 수 있음 |
| OnChangedReturnDelay | uiReturnDelayTime의 변경된 값을 연결된 ControlItem에 반영한다.  uiReturnDelayTime Event와 연결하여 사용할 수 있음 |
| OnChangedReturnLevel | uiReturnLevel의 변경된 값을 연결된 ControlItem에 반영한다.  uiReturnLevel Event와 연결하여 사용할 수 있음 |

Details

* ControlUIInfo의 UI parameter 포맷
  + 현재 param은 사용하지 않음
  + Version 1
    - Baudrate의 uiBaudrate사용
  + Version2
    - Baudrate의 uiBaudrate2 사용

# AlramUI

ControlUIInfo 중 Alram 관련 내용에 대응하기 위한 Class

의존성

* ControlUI
* UGUI

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| uiVoltageShutdown | ControlItem 중 AlramShutdown의 Voltage flag를 설정하기 위한 GUI |
| uiAngleShutdown | ControlItem 중 AlramShutdown의 Angle flag를 설정하기 위한 GUI |
| uiOverheatingShutdown | ControlItem 중 AlramShutdown의 Overheating flag를 설정하기 위한 GUI |
| uiRangeShutdown | ControlItem 중 AlramShutdown의 Range flag를 설정하기 위한 GUI |
| uiChecksumShutdown | ControlItem 중 AlramShutdown의 Checksum flag를 설정하기 위한 GUI |
| uiOverloadShutdown | ControlItem 중 AlramShutdown의 Overload flag를 설정하기 위한 GUI |
| uiInstructionShutdown | ControlItem 중 AlramShutdown의 Instruction flag를 설정하기 위한 GUI |
| uiTitleAlramLED | AlramLED를 사용하지 않는 Version의 경우 Show/Hide설정을 위한 GUI |
| uiVoltageLED | ControlItem 중 AlramLED의 Voltage flag를 설정하기 위한 GUI |
| uiAngleLED | ControlItem 중 AlramLED의 Angle flag를 설정하기 위한 GUI |
| uiOverheatingLED | ControlItem 중 AlramLED의 Overheating flag를 설정하기 위한 GUI |
| uiRangeLED | ControlItem 중 AlramLED의 Range flag를 설정하기 위한 GUI |
| uiChecksumLED | ControlItem 중 AlramLED의 Checksum flag를 설정하기 위한 GUI |
| uiOverloadLED | ControlItem 중 AlramLED의 Overload flag를 설정하기 위한 GUI |
| uiInstructionLED | ControlItem 중 AlramLED의 Instruction flag를 설정하기 위한 GUI |
| uiLED | ControlItem 중 LED를 설정하기 위한 GUI |
| uiLEDColor | ControlItem 중 ColorLED를 설정하기 위한 GUI |
| uiLEDColorText | ColorLED를 사용하지 않는 Version의 경우 Show/Hide설정을 위한 GUI |
| uiReset | ControlItem 항목을 default value로 바꾸기 위한 GUI |
| uiSave | ControlItem 변경된 항목을 CommProduct에 전송하기 위한 GUI |
| uiLowVoltage | ControlItem 중 VoltageLowLimit을 설정하기 위한 GUI |
| uiCurrentVoltage | ControlItem 중 Voltage을 설정하기 위한 GUI |
| uiHighVoltage | ControlItem 중 VoltagHighLimit을 설정하기 위한 GUI |
| uiLowVoltageInput | ControlItem 중 VoltageLowLimit을 설정하기 위한 GUI |
| uiHighVoltageInput | ControlItem 중 VoltagHighLimit을 설정하기 위한 GUI |
| uiLowVoltageValue | ControlItem 중 VoltageLowLimit의 값을 Unit으로 표현하기 위한 GUI |
| uiCurrentVoltageValue | ControlItem 중 Voltage의 값을 Unit으로 표현하기 위한 GUI |
| uiHighVoltageValue | ControlItem 중 VoltageHighLimit의 값을 Unit으로 표현하기 위한 GUI |
| uiCurrentTemperature | ControlItem 중 Temperature의 값을 Unit으로 표현하기 위한 GUI |
| uiHighTemperature | ControlItem 중 TemperatureHighLimit의 값을 설정하기 위한 GUI |
| uiHighTemperatureInput | ControlItem 중 TemperatureHighLimit의 값을 설정하기 위한 GUI |
| uiCurrentTemperatureValue | ControlItem 중 Temperature의 값을 Unit으로 표현하기 위한 GUI |
| uiHighTemperatureValue | ControlItem 중 TemperatureHighLimit의 값을 Unit으로 표현하기 위한 GUI |

Details

* ControlUIInfo의 UI parameter 포맷
  + 현재 param은 사용하지 않음
  + Version 1
    - AlramLED 사용
    - Toggle 방식 LED GUI
  + Version2
    - AlramLED 사용 안함
    - Color선택 방식 LED GUI

# AuxDeviceUI

ControlUIInfo 중 AuxDevice 관련 내용에 대응하기 위한 Class

의존성

* ControlUI
* UGUI

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| uiPorts | UI param중 port정보를 display하기 위한 GUI |
| uiDevices | 선택한 포트의 AuxDevice 리스트를 보여주기 위한 GUI |
| uiDevice | uiDevices에 ListItem을 넣기위한 Template |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| OnChangedPort | uiPorts의 선택이 바뀌었을 때 uiDevices의 내용을 바꾼다.  uiPorts의 이벤트와 연결하여 사용할 수 있음 |

Details

* ControlUIInfo의 UI parameter 포맷
  + 포트 별 정보 구분자: ‘ , ’
  + 포트 번호와 사용 가능한 Device목록 구분자: ‘ [ ‘ ‘] ‘
  + Device 목록 구분자: ‘ | ‘
  + Device Add/Subtract 결정 기호: ‘ – ‘
  + 미리 정의된 Device 이름
    - AuxMotor
    - AuxServo
    - AuxIR
    - AuxDMS
    - AuxTouch
    - AuxLED
    - AuxTemperature
    - AuxUltraSonic
    - AuxMagnetic
    - AuxMotion
    - AuxColor
    - AuxCustom
  + Version 은 현재 1만 존재

# DriveUI

ControlUIInfo 중 Drive 관련 내용에 대응하기 위한 Class

의존성

* ControlUI
* UpdownValue
* DialControl
* UGUI

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| uiJointMode | Joint Mode를 설정하기 위한 GUI |
| uiWheelMode | Wheel Mode를 설정하기 위한 GUI |
| uiJointView | Joint Mode와 관련된 GUI를 담고있는 Root |
| uiWheelView | Wheel Mode와 관련된 GUI를 담고있는 Root |
| uiSave | ControlItem 중 Saveable 속성을 갖고 있는 값을 전송하기 위한 GUI |
| uiMoving | ControlItem 중 Moving의 값을 표현하기 위한 GUI |
| uiCWPresentLoad | ControlItem 중 PresentLoad의 CW방향 값을 표현하기 위한 GUI |
| uiCCWPresentLoad | ControlItem 중 PresentLoad의 CCW방향 값을 표현하기 위한 GUI |
| uiPresentLoadValue | ControlItem 중 PresentLoad의 값을 Unit으로 표현하기 위한 GUI |
| uiCWLimitDial | ControlItem 중 CWLimit의 값을 설정하기 위한 GUI |
| uiCCWLimitDial | ControlItem 중 CCWLimit의 값을 설정하기 위한 GUI |
| uiGoalPosDial | ControlItem 중 GoalPos의 값을 설정하기 위한 GUI |
| uiPresentPosDial | ControlItem 중 PresentPos의 값을 설정하기 위한 GUI |
| uiDialKnob | Dial방식으로 CWLimit, CCWLimit, GoalPos를 설정하기 위한 GUI |
| uiAngleInput | CWLimit, CCWLimit, GoalPos를 설정하기 위한 GUI |
| uiAngleCenter | Angle의 Center값을 설정하기 위한 GUI |
| uiCWLimitSelector | uiDialKnob과 uiAngleInput을 CWLimit용으로 선택하기 위한 GUI |
| uiCCWLimitSelector | uiDialKnob과 uiAngleInput을 CCWLimit용으로 선택하기 위한 GUI |
| uiGoalPosSelector | uiDialKnob과 uiAngleInput을 GoalPos용으로 선택하기 위한 GUI |
| uiPresentPosValue | ControlItem 중 PresentPos 값을 Unit으로 표현하기 위한 GUI |
| uiCWLimitValue | ControlItem 중 CWLimit 값을 Unit으로 표현하기 위한 GUI |
| uiCCWLimitValue | ControlItem 중 CCWLimit 값을 Unit으로 표현하기 위한 GUI |
| uiGoalPosValue | ControlItem 중 GoalPos 값을 Unit으로 표현하기 위한 GUI |
| uiJointSpeedSlider | ControlItem 중 JointSpeed를 설정하기 위한 GUI |
| uiPresentJointSpeedSlider | ControlItem 중 PresentJointSpeed를 설정하기 위한 GUI |
| uiJointSpeedInput | ControlItem 중 JointSpeed의 값을 설정하기 위한 GUI |
| uiPresentJointSpeedValue | ControlItem 중 PresentJointSpeed의 값을 Unit으로 표현하기 위한 GUI |
| uiWheelSpeedSlider | ControlItem 중 WheelSpeed의 값을 설정하기 위한 GUI |
| uiPresentWheelSpeedSlider | ControlItem 중 PresentWheelSpeed의 값을 설정하기 위한 GUI |
| uiWheelSpeedInput | ControlItem 중 WheelSpeed의 값을 설정하기 위한 GUI |
| uiWheelSpeedValue | ControlItem 중 WheelSpeed의 값을 Unit으로 표현하기 위한 GUI |
| uiPresentWheelSpeedValue | ControlItem 중 PresentWheelSpeed의 값을 Unit으로 표현하기 위한 GUI |
| uiWheelSpeedStop | ControlItem 중 WheelSpeed의 값을 0으로 설정하기 위한 GUI |

Details

* ControlUIInfo의 UI parameter 포맷
  + Parameter 항목 구분자: ‘ , ‘
    - Min/Max Angle Unit 구분자: ‘~’
    - Min/Max RPM Unit 구분자: ‘~’
  + Version 1 : AX-12 방식
  + Version2 : MX 시리즈 방식
    - Version1 방식 계승
    - MultiTurnOffset 추가 (구현 안됨)
    - ResolutionDivider 추가 (구현 안됨)
    - GoalAccelerator 추가 (구현 안됨)
  + Version3: XL 시리즈 방식
    - Version1 방식 계승
    - Drive Mode 추가 (구현 안됨)

# OperateUI

ControlUIInfo 중 Operate 관련 내용에 대응하기 위한 Class

의존성

* ControlUI
* UGUI

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| V1\_uiView | Version1 GUI를 표현하기 위한 Root |
| V2\_uiView | Version2 GUI를 표현하기 위한 Root |
| V2\_uiSleepTimerInput | Version2용 ControlItem중 SleepTimer를 설정하기 위한 GUI |
| V2\_uiSleepTimerUnitValue | Version2용 ControlItem중 SleepTimer를 설정하기 위한 GUI |
| V2\_uiButtonCountValue | Version2용 ControlItem중 ButtonCount를 설정하기 위한 GUI |
| V3\_uiView | Version3 GUI를 표현하기 위한 Root |
| V4\_uiView | Version4 GUI를 표현하기 위한 Root |
| V4\_uiButton | Version4용 ControlItem 중 Button을 표현하기 위한 GUI |
| V4\_uiLED | Version4용 ControlItem중 LED를 설정하기 위한 GUI |

Details

* ControlUIInfo의 UI parameter 포맷
  + Parameter 사용 안함
  + Version1 : CM-530 방식 (구현 안됨)
  + Version2 : CM-150 방식
  + Version3: CM-200 방식 (구현 안됨)
  + Version4: OpenCM 방식

# TorqueUI

ControlUIInfo 중 Torque 관련 내용에 대응하기 위한 Class

의존성

* ControlUI
* UpdownValue
* UGUI

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| uiMaxTorqueInput | ControlItem 중 MaxTorque의 값을 설정하기 위한 GUI |
| uiMaxTorqueSlider | ControlItem 중 MaxTorque의 값을 설정하기 위한 GUI |
| uiMaxTorqueUnitValue | ControlItem 중 MaxTorque의 값을 Unit으로 표현하기 위한 GUI |
| uiTorqueLimitView | ControlItem 중 TorqueLimit의 값을 설정하기 위한 GUI |
| uiTorqueLimitInput | ControlItem 중 TorqueLimit의 값을 설정하기 위한 GUI |
| uiTorqueLimitSlider | ControlItem 중 TorqueLimit의 값을 설정하기 위한 GUI |
| uiTorqueLimitUnitValue | ControlItem 중 TorqueLimit의 값을 Unit으로 표현하기 위한 GUI |
| uiPunchInput | ControlItem 중 Punch의 값을 설정하기 위한 GUI |
| uiPunchSlider | ControlItem 중 Punch의 값을 설정하기 위한 GUI |
| uiPunchUnitValue | ControlItem 중 Punch의 값을 Unit으로 표현하기 위한 GUI |
| uiTorqueEnable | ControlItem 중 TorqueEnable의 값을 설정하기 위한 GUI |
| uiSave | ControlItem 중 Saveable 속성을 갖고 있는 값을 전송하기 위한 GUI |

Details

* ControlUIInfo의 UI parameter 포맷
  + Parameter 사용 안함
  + Version1 : 기본 방식
  + Version2 : XL 방식
    - TorqueLimit가 사용 안함

# RoboPlusManager

RoboPlusManager Demo의 UI를 표현하기 위한 Class

의존성

* CommSocket
* CommProtocol
* CommProductUI
* LocalizationManager
* ListView
* ListItem
* UGUI

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| Ui.Language | Language를 선택하기 위한 GUI |
| Ui.DeviceList | CommSocket의 foundDevices를 보여주기 위한 GUI |
| Ui.DeviceItem | Ui.DeiveList에서 사용할 ListItem Template |
| Ui.commShows | CommSocket관련 UI를 보여주기 위한 GUI  현재, 사용하지 않고 직접 UGUI Event에서 처리 |
| Ui.commCloses | CommSocket설정을 완료하기 위한 GUI  현재, 사용하지 않고 직접 UGUI Event에서 처리 |
| Ui.commSearch | CommSocket의 StartSearch를 수행하기 위한 GUI |
| Ui.messageRoot | CommSocket상태에 따른 Message를 보여주기 위한 Root |
| Ui.messageConnecting | Connecting Message Root |
| Ui.messageOpenFailed | OpenFailed Message Root |
| Ui.messageErrorClose | ErrorClose Message Root |
| Ui.commProduct | CommProductUI |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| ApplicationQuit | 프로그램 종료  직접 UGUI Event로 호출해야 함 |
| InitSettings | Setting관련 GUI 초기화  직접 UGUI Event로 호출해야 함  현재는 Language관련 설정만 함 |
| ApplySettings | Setting GUI 완료 시 호출  직접 UGUI Event로 호출해야 함 |
| CommSocketSearch | CommSocket StartSearch관련 GUI 수행 |
| CommSocketClose | CommSocket Close관련 GUI 수행 |
| ApplicationLoadLevel | Demo를 위해 다른 Scene을 호출 |

# CommProductUI

CommProduct의 UI를 표현하기 위한 Class

의존성

* CommProduct
* ControlUIManager
* ListView
* ListItem
* UGUI

Property

|  |  |
| --- | --- |
| 이름 | 설명 |
| prefab | CommProduct prefab |
| uiModelInfo | 선택된 CommProduct의 정보를 표현하기 위한 GUI |
| uiModelImage | 선택된 CommProduct의 이미지를 표현하기 위한 GUI |
| uiControlTable | ControlItem 리스트를 표현하기 위한 GUI |
| uiControlItem | uiControlTable 리스트에 사용될 ListItem Template |
| uiManager | ControlUIManager |
| uiFind | CommProduct Connect를 수행할 GUI |
| uiProductList | 연결된 CommProduct 리스트를 표현하기 위한 GUI |
| uiProductItem | uiProductList에 사용될 ListItem Template |
| uiMessageRoot | Message를 Display할 Root |
| uiMessageSearching | CommProduct 연결 중 Message |
| uiSearchingStatus | uiMessageSearching에 Display할 GUI |

Method

|  |  |
| --- | --- |
| 이름 | 설명 |
| SetFindChildProduct | True: ID 200으로 제어기를 찾고 ID 1부터 하위 Product 자동 검색  False: 선택한 ID만 검색 |
| SetFindEnable | uiFind를 Enable/Disable 상태로 만듬 |
| ClearProduct | 현재 연결된 CommProduct List를 모두 Clear시킴 |
| CancelFind | CommProduct Search를 멈춤 |