

TXLZClient

LZClient "Lazy" Component "Master"

Using :

"Micset" Simple Method For Read/Write Tags(Variables) From/To MCU
Automatically Scan Tags from MCU(Slave) And Send Set Command When The Value Changes.

SubObjects :

TXLZTag Tag

Events :

OnSendValue Aftrer Send Transaction

Commands :

Start Start Communication

Stop Stop Communication

Propertyes :

Delay Slow down communication for debug or low speed MCU (ms.)

TimeOut Read Time Out

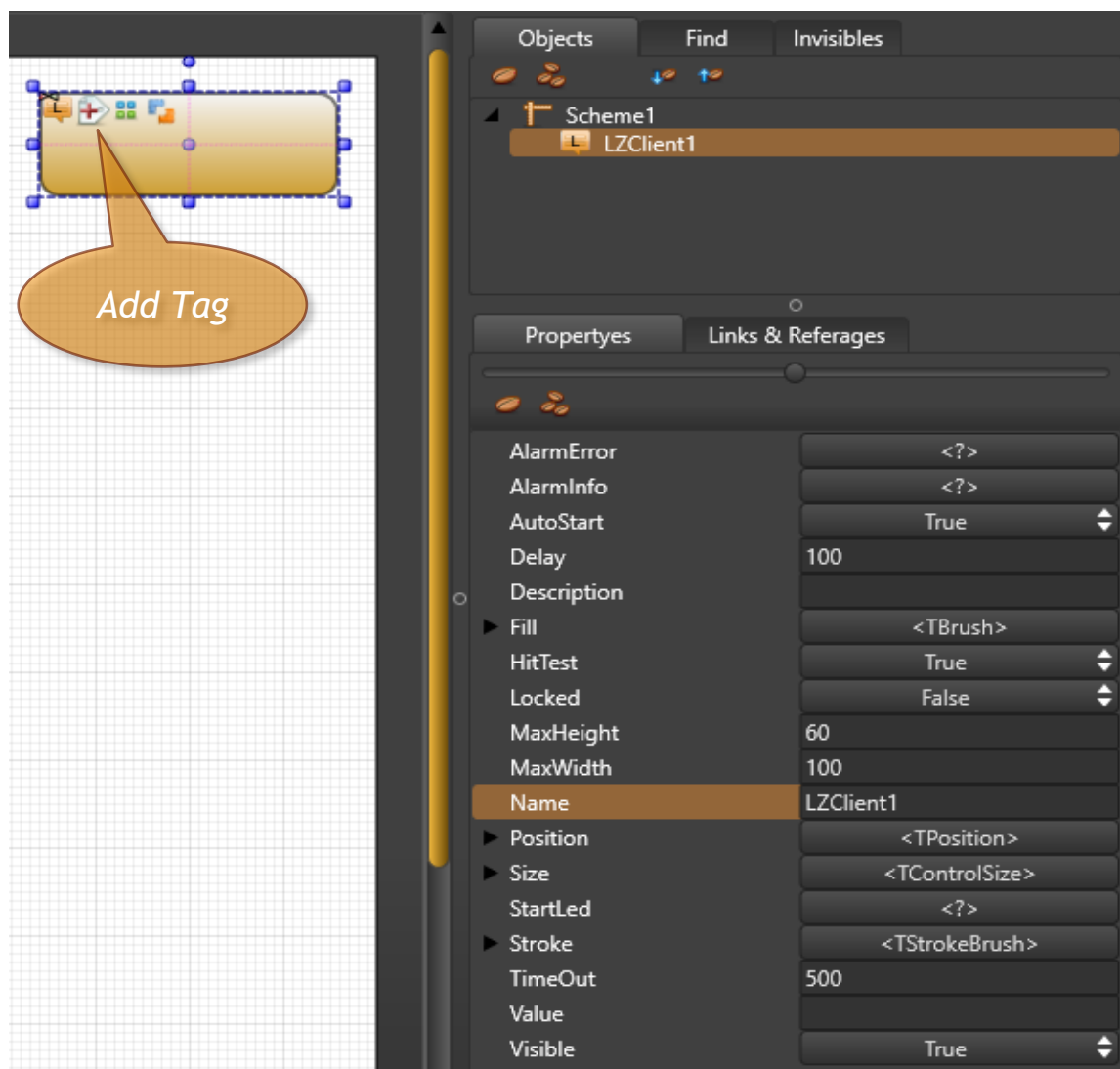
Value Link Point

Separator Separator between RemoteTagName and Value

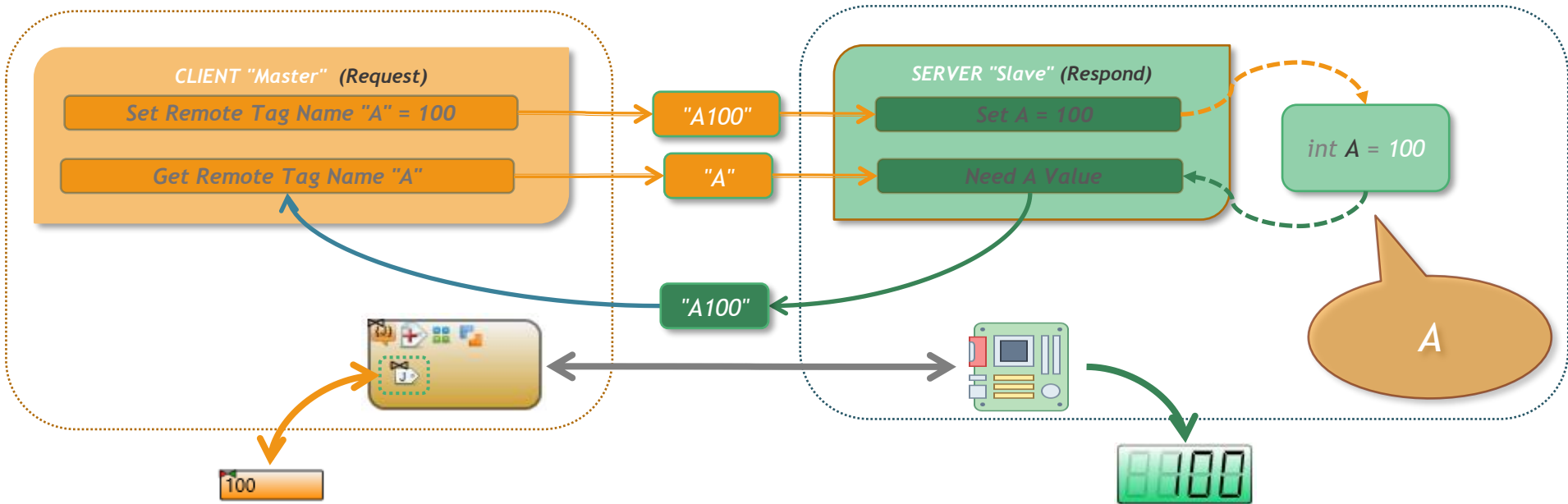
Separator = ':' , RemoteTagName = "A" TagValue = 100

"A:100" Set Value By Name A = 100

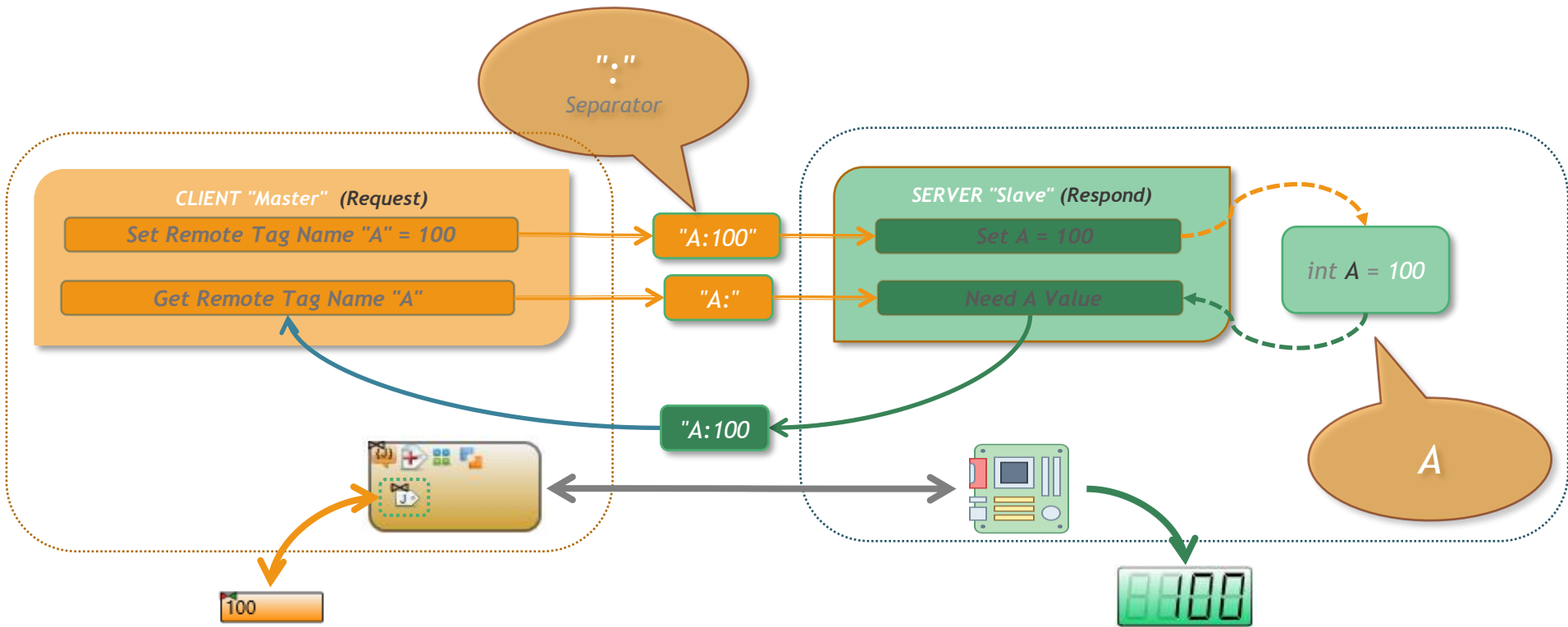
"A:" Read Value By Name A



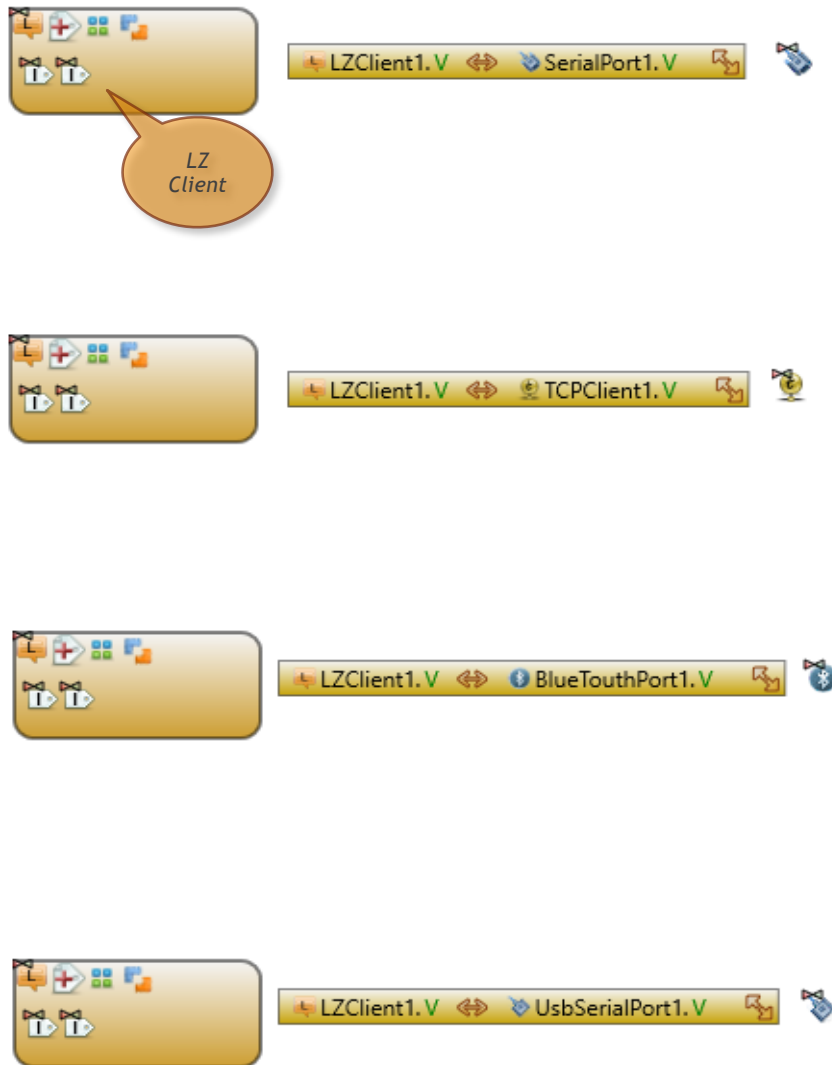
WithOut Separator



With Separator



Link



```
////////////////////////////////////
// LZ Server Erol CALISKAN    //
////////////////////////////////////
```

```
const int  MAXTAG = 6;
const int  COMMAND_LEN = 1; // A,B.. 1 char
```

```
const String LineEnd = String("\r\n");
String  StringBuf;
```

```
typedef struct TTags {
    String TagName;
    /* float*/ int TagValue;
};
```

```
TTags Tags[MAXTAG];
```

```
void ScanLZPacked(void) {
```

```
    String ValueString;
    char inChar;
    //in buffer "A1\r\n" A=1
    //in buffer "A\r\n" A=?

    if ( Serial.available() )
    {
        inChar = Serial.read(); // Read a character
        StringBuf += inChar;
        if (StringBuf.endsWith(LineEnd))
        {
            for ( int i = 0; i < MAXTAG ; i++){
                if ( StringBuf.startsWith(Tags[i].TagName )) {
                    ValueString = StringBuf.substring( COMMAND_LEN , StringBuf.length() - LineEnd.length());
                    if ( ValueString.length() > 0 ) Tags[i].TagValue = ValueString.toInt();
                    else
                        Serial.print(Tags[i].TagName + Tags[i].TagValue + LineEnd);
                    break;
                }// endif
            }//end for
            StringBuf = "";
        }// end if
    }// end while
} // end ReadFromScada
```

```
void setup() {
```

```
    // setup serial
    Serial.begin (9600);
```

```
    // define tags
    Tags[0] = (TTags) {"A",10}; // Default Value = 10 TagName = 'A' Analog 1
    Tags[1] = (TTags) {"B",20}; // Default Value = 20 TagName = 'B' Analog 2
    Tags[2] = (TTags) {"C",0};  // Default Value = 0 TagName = 'C' Button 1
    Tags[3] = (TTags) {"D",0};  // Default Value = 0 TagName = 'D' Button 2
    Tags[4] = (TTags) {"E",0};  // Default Value = 0 TagName = 'E' Led 1
    Tags[5] = (TTags) {"F",0};  // Default Value = 0 TagName = 'F' Led 2
```

```

pinMode(8, INPUT);      // set pin to input Button 1
pinMode(9, INPUT);      // set pin to input Button 2

pinMode(10, OUTPUT);    // set pin to Led 1
pinMode(11, OUTPUT);    // set pin to Led 2

pinMode(A0, INPUT);     // set pin to Analog1
pinMode(A1, INPUT);     // set pin to Analog2

}

void loop() {

  // Get analog inputs
  Tags[0].TagValue = (float)analogRead(A0); // Analog 1
  Tags[1].TagValue = (float)analogRead(A1); // Analog 1

  // Get Buttons
  Tags[2].TagValue = digitalRead(8); // Get Button 1
  Tags[3].TagValue = digitalRead(9); // Get Button 2

  // Set Leds
  if( Tags[4].TagValue == 1 ) digitalWrite(10, HIGH); else digitalWrite(10, LOW);
  if( Tags[5].TagValue == 1 ) digitalWrite(11, HIGH); else digitalWrite(11, LOW);

  ScanLZPacked();

  delay(10);

  // User Task
  // ...

}

```