Case Study for COTS Chips with UCDL

For a better indruction, we provide a set of tables that describe the heterogeneous features of 20 different chips in detail through UCDL, demonstrating that UCDL has the capability of extensive support for COTS chips. In those tables, we discuss the detailed features for each chip, contains information of their type, name, communication protocol, ID, and how many detailed functions we have achieved (full or basic). Then, for each chip, we listed the detailed functions, and gave the UCDL control sequence for one of the functions. The listed functions represent the specific functions provided by the chip (not including pin configuration and data converting that provided by UCDL). Finally, we give a demonstration of data conversion for each chip, and add necessary annotations. For the data conversion, LEGO supports functions written in strings bu ulitizing QScriptEngine[1], and we achieve complex data computing for chip outputs.

The heterogeneity characteristics of COTS chips are presented in three tables (in page 20-22). The first part (Table 1) is about different types of digital chips, which is controlled through different digital buses (*e.g.*, I2C, SPI, I2S); the second part (Table 2) is about different types of analog chips, which is controlled by ADC (analog-to-digital converter) and DAC (digital-to-analog converter) on LEGO device; the final part (Table 3) is the comparison of chips in the same function class. We provide the ID for each chip, based on which readers can easily find the original file in the anonymous open source link. It should be noted that there are a few differences in variable naming between the source file and the content discussed in the table, because we have made readability optimization in the table presentation. The details are as follows, where our UCDL gives a good support for all of them.

Table 1: **Features comparison of COTs Chips (Part 1: Digital chips in different functions)**

1	**Type: 3-Axis Digital Angular Rate Gyroscope Chip: FXAS21002 ID: 0xFAEE Protocol: SPI States: All 6 Functions Achieved**		
	Detailed Functions[1]	Control Sequence of function 1	Example of Data Conversion
	1. Read Angular Rate Register Data 2. From Standby Mode Change to Active Mode 3. From Ready Mode Change to Active Mode 4. Change to Ready Mode 5. Change to Active Mode 6. Reset Device and Change to Standby Mode	1), CW(2, L), +2us; 2), DW(3, 1, 0x81), +10us; 3), DR(5, 6, X1, X2, X3, X4, X5, X6), +60us; 4), CW(2, H), +2us;	1), DF(Y1, variable1, (Y1=X1<<8)+X2, null); 2), DF(Y2, variable2, (Y2=X3<<8)+X4, null); 3), DF(Y3, variable3, (Y3=X5<<8)+X6, null); 4), DF(Y4, X-angular, (Y4=Y1,Y1<32768; Y4=(Y1-65536),Y1>=32768), degree/s); 5), DF(Y5, Y-angular, (Y5=Y2,Y2<32768; Y5=(Y2-65536),Y2>=32768), degree/s); 6), DF(Y6, Z-angular, (Y6=Y3,Y3<32768; Y6=(Y3-65536),Y3>=32768), degree/s);
	Note[1]: The listed functions represents the specific functions provided by the chip.		**Annotation:**1)"Y4, Y5, Y6" are final data of 3-axis angular rates,respectively. 2),"Y1 to Y3" are intermediate variables. 3),"X1 to X6" are raw angular data read by Function 1.
2	**Type: Digital Microphone Chip: INMP441 ID: 0x7C60 Protocol: I2S States: 1 Basic Function Achieved (1/2 of all functions)**		
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion (None[1])
	1. Get I2S Data	1), DR(2, I2S, DIRECT), +1s;	[1]**Annotation:** The I2S output data is directly send to the gateway and decode by third-party audio software.
3	**Type: Digital Accelerometer Chip: ADXL362 ID: 0xCCFC Protocol: SPI States: 3 Basic Functions Achieved (3/5 of all)**		
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
	1. Read Acceleration Data 2. Set Standby Mode 3. Set WAKEUP Mode	1), CW(5, L),+2us; 2), DW(3,0x0B),+2us; 3), DW(3,0x08),+2us; 4), DR(4,1,X1),+6us; 5), CW(5, H),+2us;	1), DF(Y1, X-axis Acceleration, (Y1=X1/64, X1<128; Y1=(X1-256)/64, X1>=128), g); **Annotation:** 1), "Y1" is the final X-axis acceleration data. 2), "X1" is an 8bit signed raw data read by Function 1
4	**Type: 256Bytes EEPROM (Memory) Chip: AT24C02 ID: 0x94B5 Protocol: I2C States: 3 Basic Functions Achieved (3/5 of all)**		
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
	1. Read All Data 2. Set Address 3. Write Data	1), DR(5,0xA1,0x00,256), +100ms;	**Annotation:** The output data of Memory is displayed directly on the gateway. (not need to make the conversion)
5	**Type: 8-bit Temperature Sensor Chip: LM99 ID: 0x8169 Protocol: SmBus States: 2 Basic Functions Achieved (2/3 of all)**		
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
	1. Read Local Temperature Register Data 2. Read Remote Temperature Register Data	1), DR(5, 0x99, 0x00, 1, X1), +64ms;	1), DF(Y1, Temperature, (Y1=X1,X1<128; Y1=X1-256,X1>=256), degree C); **Annotation:** 1), "X1" is a 8bit signed raw data. 2), "Y1" is the final temperature data.

[1]The site of QScriptEngine: https://doc.qt.io/qt-5/qscriptengine.html

▷ **Continue with table1 (Part1: Digital chips in different functions)**

6	**Type: Digital Baroceptor Chip: 2SMPB-02E ID: 0x010F Protocol: SPI States: 3 Basic Functions Achieved (3/4 of all)**		
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
	1. Read Compensation Variables 2. Read Temperature Register Data 3. Read Pressure Register Data	1), CW(2, L),+2us; 2), DW(3, 1, 0xA0),+2us; 3), DR(5, 25),+26us; 4), CW(2, H),+2us;	1), DF(Y1, variable1, Y1=X29*pow(2, 16)+X30* pow(2, 8)+X31, null); 2), DF(Y2, variable2, Y2=(X19*pow(2, 12)+X20* pow(2, 4)+X25&0x0F)/16, null); 3), DF(Y3, variable3, Y3=-6.3*pow(10, -3)+4.3* pow(10, -4)*(X21*pow(2, 8)+X22)/32767, null); 4), DF(Y4, variable4, Y4=-1.9*pow(10, -11)+1.2* pow(10, -10)*(X23*pow(2, 8)+X24)/32767, null); 5), DF(Y5, coe5, Y5=Y2+Y3*Y1+Y4*pow(Y1, 2), null); 6), DF(Y6, Temperature, Y6=Y5/256, degree C);
	Note: We present some special chips with complex data conversion that contain a large amount of compensation from different "read" functions. It indicates that our UCDL can support complex data computing for chip outputs.		**Annotation:** 1),"X19, X20, X21, X22, X23, X24, X25" are compensation variables read by Function 1. 2),"Y1-Y5" are intermediate variables. 3),"X29-X31" are raw temperature data read by Function 2. 4), Y6 is the final temperature data.

7	**Type: Digital LFV[2] Sensor Chip: SLF3S-0600F ID: 0x3938 Protocol: I2C States: 2 Basic Functions Achieved (2/3 of all)**		
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
	1. Read Flow Data 2. Soft Reset;	1), DW(1, 0x10, 0x36, 1, 0x08),+600us; 2), DR(1, NOADDR, 0x11, 3, X1, X2, X3),+100us;	1), DF(Y1, Flow Velocity, Y1=X1<<8+X2, ul/min); 2), DF(Y2, CRC, Y2=X3, null);
	Note[2]**:** LFV is the abbreviation of Liquid Flow Velocity.		**Annotation:** 1), "X1" and "X2" are the upper and lower 8-bits of the raw data for fluid velocity sensing, respectively. 2), "X3" is the CRC (Cyclic redundancy Check) code of "X1" and "X2", which helps the gateway to check the correctness of the uploaded data. 3), "Y1" is the final data for liquid flow velocity sensing.

Table 2: **Detailed features of COTs Chips (Part 2: Analog chips in different functions)**

8	**Type: Dual Axis Tilt Sensor Chip: AXISENSE-2 ID: 0xF038 Protocol: Analog Output States: All 1 Function Achieved**		
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
	1. Read Analog Angular (Achieved by ADC Sampling)	1), DR(3, ADC,X1), +10us; 2), DR(4, ADC,X2), +3ms;	1), DF(Y1, X_Axis, Y1=((5/pow(2,8))*(X1)-2.5)*45, Degree); 2), DF(Y2, Y_Axis, Y2=((5/pow(2,8))*(X2)-2.5)*45, Degree); **Annotation:** 1), "X1, X2" are the raw data output by ADC. 2), "X1, X2" are the final data for tilt angle in X-axis and Y-axis, respectively
9	**Type: UV (ultraviolet ray) Sensor Chip: ML8511A ID: 0x7DB9 Protocol: Analog Output States: All 1 Function Achieved**		
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
	1. Read Analog UV (Ultraviolet Ray) Value (Achieved by ADC Sampling)	1), DR(4, ADC,X1), +1ms;	1), DF(Y1, UV Intensity,Y1=((5/pow(2,8))*(X1)-1)*7.5, mW/cm2); **Annotation:** 1), "X1" is the raw data output by ADC. 2), "Y1" is the final data for UV intensity sensing
10	**Type: Analog Dust Sensor Chip: GP2Y1014AU ID: 0x126F Protocol: Analog Output States: All 1 Function Achieved**		
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
	1. Data Reading (Achieved by ADC Sampling)	1), CW(3, H), +280us; 2), DR(5, ADC, X1), +40us; 3), CW(3, L), +9680us;	1), DF(Y1, Dust-density, Y1=0.17*X1*5/255-0.1, mg/m^3); **Annotation:** 1), "X1" is the raw data output by ADC. 2), "Y1" is the final data for dust-density sensing.
11	**Type: Analog Alcohol Sensor Chip: MQ3 ID: 0x39ED Protocol: Analog Output States: All 3 Functions Achieved**		
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
	1. Get Raw Data (Achieved by ADC Sampling) 2. MQ3 warm up 3. Check if alcohol exists	1), DR(4, ADC, X1), +3ms;	1), DF(Y1, Y1=rawData, X1/255*5, null); 2), DF(Y2, Alcohol, Y2=(Y1-1)*20, ppm); **Annotation:**1), "X1" is the raw data output by ADC. 2), "Y1" is the analog voltage output by the chip. 3), "Y2" is the final data for alcohol concentration sensing

▷ **Continue with table2 (Part2: Analog chips in different functions)**

12	**Type: Analog Gas Sensor Chip: MQ5 ID: 0x454B Protocol: Analog Output States: All 3 Functions Achieved**		
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
	1. Get Raw Data (Achieved by ADC Sampling) 2. MQ5 warm up 3. Check if gas exists	1,) DR(4, ADC, X1), +3ms;	1,) DF(Y1, rawData, Y1=X1/255*5, null); 2,) DF(Y2, gas concentration, Y2=(Y1-1)*10*20, ppm);
			Annotation: 1,) "X1" is the raw data output by ADC. 2,) "Y1" is the analog voltage output by the chip. 3), "Y2" is the final data for gas concentration sensing
13	**Type: Analog AMR[3] Sensor Chip: ADA4571 ID: 0xC790 Protocol: Analog Output States: All 1 Function Achieved**		
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
	1. Read Analog Output (achieved by ADC sampling)	1), DR(2, ADC,X1), +10us; 2), (4, ADC,X2), +3ms;	1), DF(Y1, variable1, Y1=X1, null); 2), DF(Y2, variable2, Y2=X2, null); DF(Y3, Angle, arctan(Y2/Y1)/2, Degree);
	Note[3]: The function of this chip is AMR (An-isotropic Magneto Resistivity) angular sensing.		**Annotation:** 1), "X1, X2" are raw data output by ADC, 2), "Y1,Y2" are the intermediate variables in the data conversion. 3), "Y3" is the final data for angle sensing.
14	**Type: Analog Rotary Position Sensor Chip: AMS22S ID: 0x1032 Protocol: Analog Output States: All 1 Function Achieved**		
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
	1. Read Analog Angular (Achieved by ADC Sampling)	1), DR(2, ADC,X1), +3ms;	1), DF(Y1, Angle, Y1=((5/pow(2,8))*(X1))/5*360, Degree);
			Annotation: 1), "X1" is the raw data output by ADC. 2), Y1 is the final data for angular position sensing

Table 3: **Detailed features of COTs Chips (Part 3: COTS chips in same function classes)**

15	**Type: Ambient Light Sensor Chip: MAX9635 ID: 0x7B31 Protocol: I2C States: 2 Basic Functions Achieved (2/3 of all)**		
	Detailed Functions	Control Sequence of Function 1	Example of Data Conversion
	1. Read Light Register Data in Continuous Mode 2. Config Continous Mode	1), DR(5, 0x95, 0x03, 2, X1, X2),+7ms;	1), DF(Y1, variable1, Y1=(X1&0xF0)>>4, null); 2) DF(Y2, variable2, ((X1&0x0F)<<4)+(X2&0x0F), null); 3), DF(Y3, Lux, pow(2, EXP)*Y2*0.045, lux);
	Note: We have added some chips with similar functions. It indicates that COTS chips have heterogeneous characteristics even in the same function classes, and our UCDL can describe their features well.		**Annotation:**1), "X1" and "X2" are the raw data readed in Function 1 2), "Y1" and "Y2" are the intermediate variable in data conversion 3), "Y3" is the final data for ambient light sensing
16	**Type: Ambient Light Sensor Chip: TSL2562 ID: 0xC771 Protocol: SMBus States: 3 Basic Functions Achieved (3/4 of all)**		
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
	1. Read Register Data 2. Set Sensor Power down 3. Set Sensor Power up	1), DR(5, 0x53, 0xAC, 2, X1, X2), +100us; 2), DR(5, 0x53, 0xAE, 2, X3, X4), +400ms;	1), DF(Y1, variable1, Y1=X1+X2<<8, null); 2), DF(Y2, intermediate variable, Y2=X3+X4<<8, null); 3), DF(Y3, intermediate variable,Y3=Y2/Y1, null); 4), DF(Y4, Lux, (Y4=0.0315*Y1-0.0593*Y1*pow(CH10, 1.4),0<Y3<=0.52; Y4=0.0229*CH0-0.0291*Y2,0.52<Y3<=0.65; Y4=0.0157*Y1-0.0180*Y2,0.65<Y3<=0.80; Y4=0.00338*Y1-0.00260*Y2,0.80<Y3<=1.3; Y4=0,Y3>1.3), degree C);
	Note: The data conversion for this chip is relatively complex, but we can disassemble the process into multiple DF statements with intermediate variables. It proves that our UCDL can support complex computing for chip outputs effectively.		**Annotation:** 1), "X1-X4" are 8 bit raw data read by Function1, 2), "Y1,Y2,Y3" are intermediate variable of the data converting, 3), "Y4" is the final data for ambient light sensing.

▷ **Continue with 3 (Part3: COTS chips in same function classes)**

	Type: Relative Pressure Sensor	**Chip: MLX90809 ID:0xF921**	**Protocol: SENT States: All 1 Function Achieved**
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
17	1. Get Raw Data	1), DR(2, 3, X1, X2, X3), +1ms;	1), DF(Y1, variable1, X1<<8+X2<<4+X3, null); 2), DF(Y2, Pressure, (Y2=Y1/4096*3, Y1<2048; Y2=(Y1-4096)/4096*3-1.5, Y1>=2048), %FSO);
			Annotation: 1) "X1-X3" is the raw data readed by Function 1 2) "Y1" is the intermediate variable 3) "Y2" is the final data for relative pressure sensing

	Type:Digital Pressure Sensor	**Chip: MS5525DSO ID: 0xE52C**	**Protocol: SPI States: 4 Basic Functions Achieved (4/6 of all)**
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
18	1. Read Pressure Register Data 2. Read Temprature Register Data 3. Read Calibration Data 4. Reset	1), CW(4, L),+2us; 2), DW(2, 1, 0x48),+9ms; 3), CW(4, H),+2us; 4), CW(4, L),+2us; 5), DW(2, 1, 0x00),+2us; 6), DR(1, 3, X13, X14, X15),+1us; 7), CW(4, H),+2us;	1), DF(Y11, variable1, Y11=C2*pow(2, 17)+(C3*dT)/128, null); 2), DF(Y12, variable2, Y12=C1*pow(2, 15)+(C3*dT)/128, null); 3), DF(Y13, Pressure, (D1*Y12/pow(2, 21)-Y11)/pow(2, 15), psi); **Annotation:** 1), "C1, C2, C3, dT" are computation results of other "DF" statements. 2), "Y11,Y12" are intermediate variables 3), "Y13" is the final data for air pressure sensing.

	Type: Digital Temperature Sensor	**Chip: DS18B20 ID: 0xAC1B**	**Protocol: 1-Wire States: 3/5 Functions Achieved**
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
19	1. Read Temperature 2. Init SD18B20 3. Read 8 Bytes Serial Code	1), CW(2, L), +750us; 2), CW(2, H), +15us; 3), DW(2, 2, 0xCC, 0x44), +200us; 4), CW(2, L), +750us; 5), CW(2, H), +15us; 6), DW(2, 2, 0xCC, 0xBE), +200us; 7), DR(2, 2, X1, X2), +200us; composed of X1 and X2.	1), DF(Y1, RawData, X1+X2<<8, null); 2), DF(Y2, Temp, (y=RawData*0.0625, RawData<32768; y=(RawData-65536)*0.0625, RawData>=32768), degree); **Annotation:** 1), "X1,X2" are the upper and lower 8-bits raw temperature data. 2), "Y1" is the 16-bit of raw temperature data. 3), "Y2" is the final temperature data.

	Type: Temperature and RH Sensor	**Chip: ENS210 ID: 0xB1D2**	**Protocol: I2C States: All 10 Functions Achieved**
	Detailed Functions	Control Sequence of function 1	Example of Data Conversion
20	1. Read Temperature Registera 2. Disable LowPower Mode 3. Enable LowPower Mode 4. Temp & RH Continuous Mode, 5. Temperature Continuous Mode, RH Single Shot Mode 6. Temperature Single Shot Mode, RH Continuous Mode 7. Temperature Single Shot Mode, RH Single Shot Mode 8. Read PART_ID and UID 9. Read RH Register Data 10. Stop Continuous Measurement	1), DW(3, 0x86, 0x22, 1, 0x01), +130ms; 2), DR(3, 0x87, 0x30, 3, X13, X14, X15), +1ms;	1), DF(Y4, TinK, Y4=(X14*pow(2, 8) +X15)/64, K); 2), DF(Y5, TinC, Y5=Y4-273.15, C); 3), DF(Y6, TinF, Y6=Y5*1.8+32.0, F); **Annotation:** 1), "X13,X14,X15" is the raw data readed from function 1 2), "Y4 (TinK)" is the Kelvin temperature. 3), "Y5 (TinC)" is the Celsius temperature. 4), "Y5 (TinF)" is the Fahrenheit temperature.
		Note: RH is the abbreviation of relative humidity. We add this chip with many functions to illustrate that the UCDL can support COTS chips with complex functions. In this case, we have implemented all ten functions.	