InME Python API

Wayne Chiu, Digital Content Laboratory, National Donghwa University

2016/09/02 Version 1

Contents

System Requirements

Hardware Requirement:

- Processor:
  - x86 Architecture Compatible
  - x86_64 Architecture Compatible
  - ARMv7 Architecture Compatible
  - ARMv8 Architecture Compatible
- Devices:
  - Bluetooth Adapter with version 3.0 or higher support

Software Requirement:

- Operating System:
  - Common Linux with Kernel 2.6 or higher
  - Recommend Linux Distro:
    - Debian-based Distro
    - RHEL/Fedora-based Distro
- Required Linux Library

Library Name	Pkg in Debian-based	Pkg in RH-Based	Other Linux
GNU Compiler Collection 4.x / 5.x	gcc	gcc	gcc
GNU Compiler Collection C++ Ext.	g++	gcc-c++	g++
Python 2	python	python	python
Python 2 Development Addon	python-dev	python-devel	Click Here
Bluez	bluez	bluez-libs	Click Here
Bluez Development Addon	libbluetooth-dev	bluez-libs-devel	Click Here
Linux Kernel Header *1	linux-headers-generic	kernel-headers	Click Here
RPM Config *2	N/A	redhat-rpm-config	N/A

*1 In some distro, installation of pybluez may failed due to the lack of the kernel header.

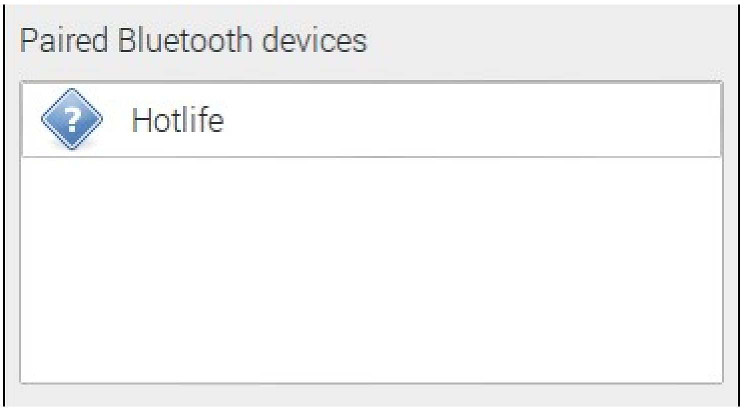
*2 In later version of Fedora (23, 24), the package RPM config, which is needed but not installed by default.

- Python Requirement:
    - Version: 2.x
    - Required Library:
        - pybluez

Getting Started with inME

Before any operation started. InME must be paired with the device or system. Otherwise, rest of the operation is not executable:

The device will be labeled as “Hotlife”，use the system bluetooth manager to be paired with inME.



Paired Bluetooth devices

Hotlife

If the pairing process requires any passcode, the passcode is 0000

Classes

Major Classes:

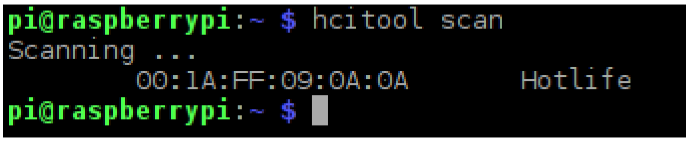
- InMEReader
- ADS1229_Descriptive_DataStruct
- ADS1229_Channel_DataStruct
- ADS1229_LeadOff_DataStruct
- AFE4490_Descriptive_DataStruct
- AFE4490_Channel_DataStruct

Class: InMEReader

Constructor:

Name:	Return:
InMEReader()	void
Caption:	
Default Constructor. No variables required.	
Example:	
InME = InMEReader()	

Member Function:

<table>
<tr><td>Name:</td><td>Return:</td></tr>
<tr><td>setInMEAddress(<i>InMEAddress&lt;str&gt;</i>)</td><td>void</td></tr>
<tr><td colspan="2">Input Variables:</td></tr>
<tr><td><i>InMEAddress&lt;str&gt;</i></td><td>Any valid MAC address format is accepted:
E.g. “00:1A:FF:09:0A:0A”</td></tr>
<tr><td colspan="2">Caption:</td></tr>
<tr><td colspan="2">Setting up inME bluetooth Address. Since each inME device has different MAC Address, it is needed to be specified before all operation. If you are not sure about your inME device bluetooth address, please execute the command:
“hcitool scan”
A device labeled with:
“hotlife”
Should be appeared on your screen, with specified MAC address:
<pre>pi@raspberrypi:~ $ hcitool scan
Scanning ...
        00:1A:FF:09:0A:0A   Hotlife
pi@raspberrypi:~ $</pre></td></tr>
<tr><td colspan="2">Example:</td></tr>
<tr><td colspan="2">InME.setInMEAddress(“00:1A:FF:09:0A:0A”)</td></tr>
</table>

Name:	Return:
connectToInME()	void
Caption:	
Establish the connection with inME	
Example:	
InME.connectToInME()	

Name:	Return:
disconnectFromInME()	void
Caption:	
Terminate the connection with inME	
Example:	
InME.disconnectFromInME()	

Name:	Return:
nextRecord()	void
Caption:	
Get a whole data set from inME	
Example:	
InME.nextRecord()	

Name:	Return:
getADS1229Descriptive()	ADS1229_Descriptive_DataStruct
Caption:	
Parse ADS1229 Descriptive Data from the local buffer. Note: nextRecord() should be performed before using this function.	
Example:	
ADS1229DD = InME.getADS1229Descriptive()	

Name:	Return:
getADS1229Channel()	ADS1229_Channel_DataStruct
Caption:	
Parse ADS1229 Channel Data from the local buffer. Note: nextRecord() should be performed before using this function.	
Example:	
ADS1229CD = InME.getADS1229Channel()	

Name:	Return:
getADS1229LeadOff()	ADS1229_LeadOff_DataStruct
Caption:	
Parse ADS1229 Lead Off Data from the local buffer. Note: nextRecord() should be performed before using this function.	
Example:	
ADS1229LO = InME.getADS1229LeadOff()	

Name:	Return:
getAFE4490Descriptive()	AFE4490_Descriptive_DataSruct
Caption:	
Parse AFE4490 Descriptive Data from the local buffer. Note: nextRecord() should be performed before using this function.	
Example:	
AFE4490DD = InME.getAFE4490Descriptive()	

Name:	Return:
getAFE4490Channel()	AFE4490_Channel_DataStruct
Caption:	
Parse AFE4490 Channel Data from the local buffer. Note: nextRecord() should be performed before using this function.	
Example:	
AFE4490CD = InME.getAFE4490Channel()	

Class: ADS1229_Descriptive_DataStruct

Constructor

Name:	Return:
ADS1229_Descriptive_DataStruct(*pkt<byte>[]*)	void
Input Variables:	
pkt<byte>[]	Bytes including ADS1229 Descriptive Info. The *pkt<byte>[]* length should be 8 bytes long.
Caption:	
Given bytes that contains ADS1229 Descriptive Information, and create an ADS1229_Descriptive_DataStruct data node.	
Example:	
ADS1229DD = ADS1229_Descriptive_DataStruct(byteArr[3:11])	

Member Variables

Name:	Data Type:
FrameID	<int>
Caption:	
Retrieve FrameID Value from ADS1229_Descriptive_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
frameID = ADS1229DD.FrameID	

Name:	Data Type:
TimeTick	<int>
Caption:	
Retrieve TimeTick Value from ADS1229_Descriptive_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
timeTick = ADS1229DD.TimeTick	

Class: ADS1229_Channel_DataStruct

Constructor

Name:	Return:
ADS1229_Channel_DataStruct(*pkt<byte>[]*)	void
Input Variables:	
pkt<byte>[]	Bytes including ADS1229 Channel Data. The *pkt<byte>[]* length should be 24 bytes long.
Caption:	
Given bytes that contains ADS1229 Channel Data, and create an ADS1229_Channel_DataStruct data node.	
Example:	
ADS1229CD = ADS1229_Channel_DataStruct(byteArr[13:37])	

Member Variables

Name:	Data Type:
CH1	<int>
Caption:	
Retrieve CH1 Value from ADS1229_Channel_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH1 = ADS1229CD.CH1	

Name:	Data Type:
CH2	<int>
Caption:	
Retrieve CH2 Value from ADS1229_Channel_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH2 = ADS1229CD.CH2	

Name:	Data Type:
CH3	<int>
Caption:	
Retrieve CH3 Value from ADS1229_Channel_DataStruct Note: Class must be initialized, otherwise specific member variables is in-aceessable.	
Example:	
CH = ADS1229CD.CH3	

Name:	Data Type:
CH4	<int>
Caption:	
Retrieve CH4 Value from ADS1229_Channel_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH4 = ADS1229CD.CH4	

Name:	Data Type:
CH5	<int>
Caption:	
Retrieve CH5 Value from ADS1229_Channel_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH5 = ADS1229CD.CH5	

Name:	Data Type:
CH6	<int>
Caption:	
Retrieve CH6 Value from ADS1229_Channel_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH6 = ADS1229CD.CH6	

Name:	Data Type:
CH7	<int>
Caption:	
Retrieve CH7 Value from ADS1229_Channel_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH7 = ADS1229CD.CH7	

Name:	Data Type:
CH8	<int>
Caption:	
Retrieve CH8 Value from ADS1229_Channel_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH8 = ADS1229CD.CH8	

Class: ADS1229_LeadOff_DataStruct

Constructor

Name:	Return
ADS1229_LeadOff_DataStruct(*pkt<byte>[]*)	void
Input Variables:	
pkt<byte>[]	Bytes including ADS1229 Lead Off Data. The *pkt<byte>[]* length should be 3 bytes long.
Caption:	
Given bytes that contains ADS1229 Lead Off Data, and create an ADS1229_LeadOff_DataStruct data node.	
Example:	
ADS1229LO = ADS1229_LeadOff_DataStruct(byteArr[0:3])	

Member Variables

Name:	Data Type:
CH1	<bool>
Caption:	
Retrieve CH1 Lead Off Boolean Value from ADS1229_LeadOff_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH1 = ADS1229LO.CH1	

Name:	Data Type:
CH2	<bool>
Caption:	
Retrieve CH2 Lead Off Boolean Value from ADS1229_LeadOff_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH2 = ADS1229LO.CH2	

Name:	Data Type:
CH3	<bool>
Caption:	
Retrieve CH3 Lead Off Boolean Value from ADS1229_LeadOff_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH3 = ADS1229LO.CH3	

Name:	Data Type:
CH4	<bool>
Caption:	
Retrieve CH4 Lead Off Boolean Value from ADS1229_LeadOff_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH4 = ADS1229LO.CH4	

Name:	Data Type:
CH5	<bool>
Caption:	
Retrieve CH5 Lead Off Boolean Value from ADS1229_LeadOff_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH5 = ADS1229LO.CH5	

Name:	Data Type:
CH6	<bool>
Caption:	
Retrieve CH6 Lead Off Boolean Value from ADS1229_LeadOff_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH6 = ADS1229LO.CH6	

Name:	Data Type:
CH7	<bool>
Caption:	
Retrieve CH7 Lead Off Boolean Value from ADS1229_LeadOff_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH7 = ADS1229LO.CH7	

Name:	Data Type:
CH8	<bool>
Caption:	
Retrieve CH8 Lead Off Boolean Value from ADS1229_LeadOff_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
CH8 = ADS1229LO.CH8	

Class: AFE4490_Descriptive_DataStruct

Constructor

Name:	Return:
AFE4490_Descriptive_DataStruct(*pkt<byte>[]*)	void
Input Variables:	
pkt<byte>[]	Bytes including AFE4490 Descriptive Info. The *pkt<byte>[]* length should be 8 bytes long.
Caption:	
Given bytes that contains AFE4490 Descriptive Information, and create an AFE4490_Descriptive_DataStruct data node.	
Example:	
AFE4490DD = AFE4490_Descriptive_DataStruct(byteArr[3:11])	

Member Variables

Name:	Data Type:
FrameID	<int>
Caption:	
Retrieve FrameID Value from AFE4490_Descriptive_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
frameID = AFE4490DD.FrameID	

Name:	Data Type:
TimeTick	<int>
Caption:	
Retrieve TimeTick Value from AFE4490_Descriptive_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
timeTick = AFE4490DD.TimeTick	

Class: AFE4490_Channel_DataStruct

Constructor

Name:	Return:
AFE4490_Channel_DataStruct(*pkt<byte>[]*)	void
Input Variables:	
pkt<byte>[]	Bytes including AFE4490 Channel Data The *pkt<byte>[]* length should be 6 bytes long
Caption:	
Given bytes that contains AFE4490 Channel Data, and create an AFE4490_Channel_DataStruct data node. The byte<>[] length should be 6 bytes long.	
Example:	
AFE4490CD = AFE4490_Channel_DataStruct(byteArr[3:9])	

Member Variables

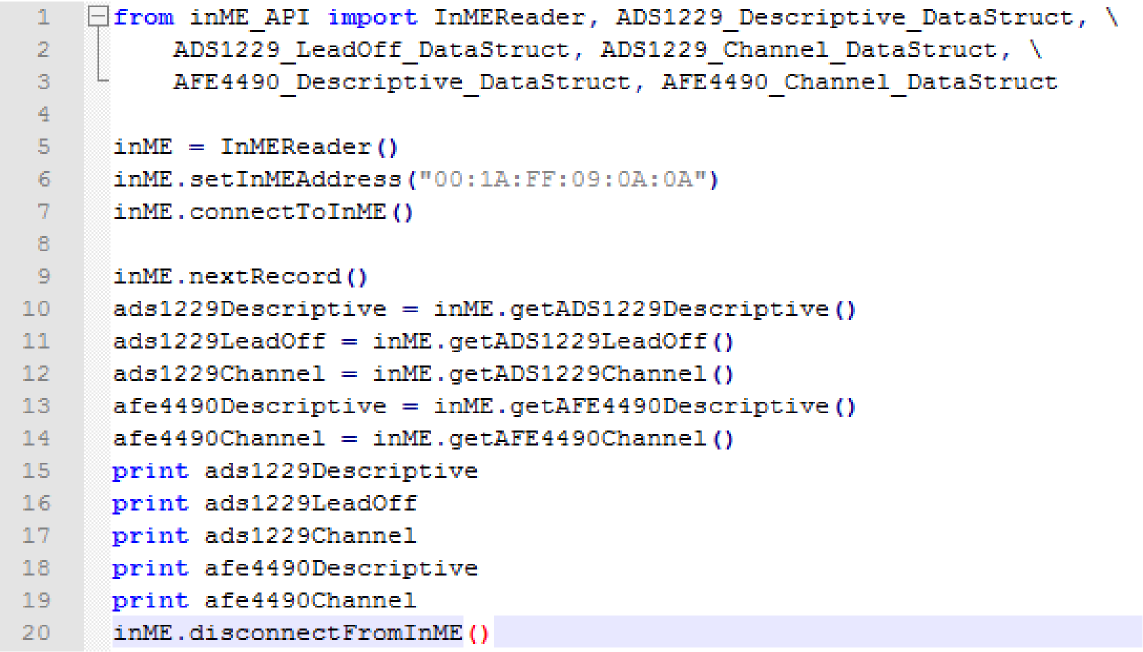
Name:	Return:
RData	Data Type: <int>
Caption:	
Retrieve Rdata Value from AFE4490_Channel_DataStruct Note: Class must be initialized, otherwise specific member variables is in-aceessable.	
Example:	
rdata = ADS1229CD.RData	

Name:	Return:
IRData	Data Type: <int>
Caption:	
Retrieve IRData Value from AFE4490_Channel_DataStruct Note: Class must be initialized, otherwise specific member variables is inaccessible.	
Example:	
irdata = ADS1229CD.IRData	

Programming Example

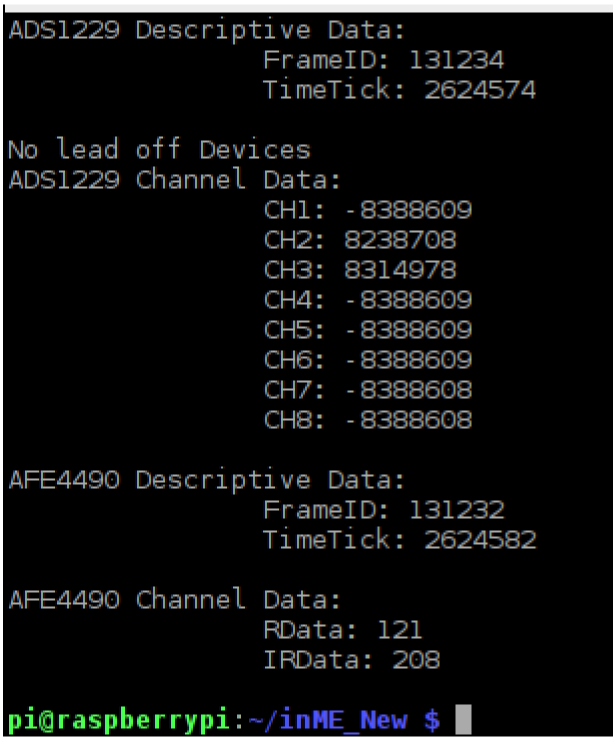
1. Simple Test Program

Code



```
from inME_API import InMEReader, ADS1229_Descriptive_DataStruct, \
    ADS1229_LeadOff_DataStruct, ADS1229_Channel_DataStruct, \
    AFE4490_Descriptive_DataStruct, AFE4490_Channel_DataStruct

inME = InMEReader()
inME.setInMEAddress("00:1A:FF:09:0A:0A")
inME.connectToInME()

inME.nextRecord()
ads1229Descriptive = inME.getADS1229Descriptive()
ads1229LeadOff = inME.getADS1229LeadOff()
ads1229Channel = inME.getADS1229Channel()
afe4490Descriptive = inME.getAFE4490Descriptive()
afe4490Channel = inME.getAFE4490Channel()
print ads1229Descriptive
print ads1229LeadOff
print ads1229Channel
print afe4490Descriptive
print afe4490Channel
inME.disconnectFromInME()
```

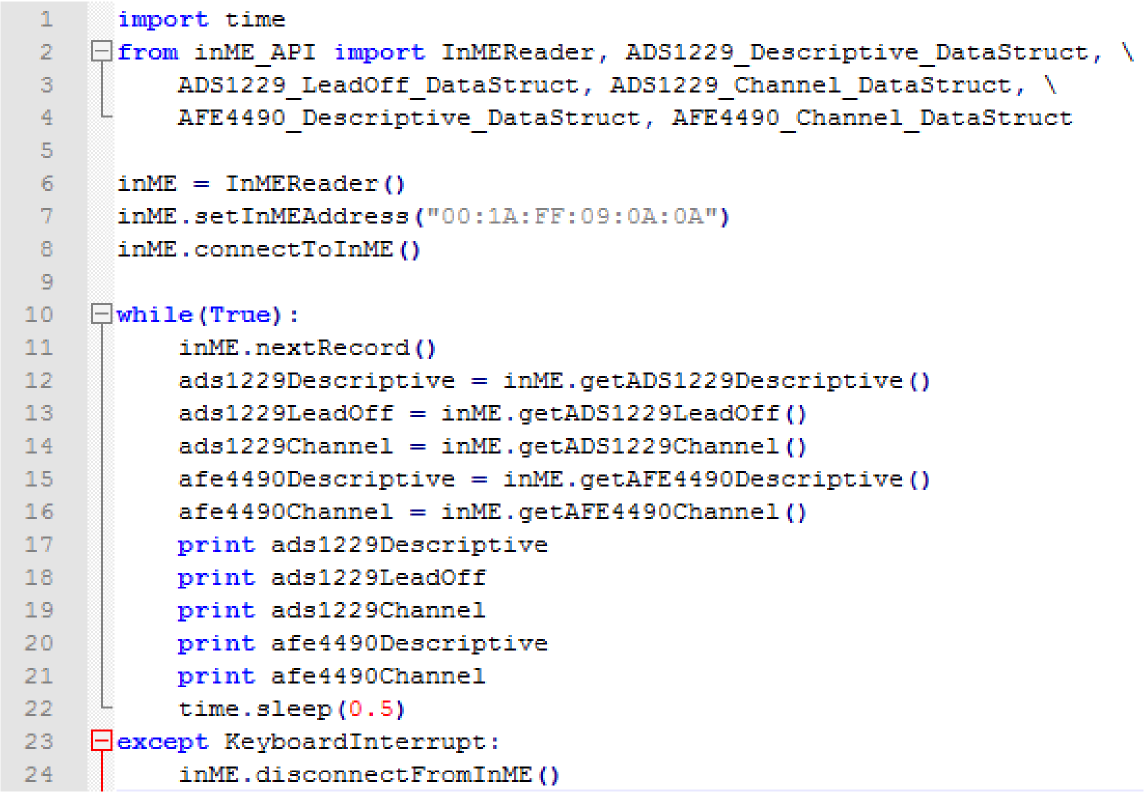
Result



```
ADS1229 Descriptive Data:
                FrameID: 131234
                TimeTick: 2624574

No lead off Devices
ADS1229 Channel Data:
                CH1: -8388609
                CH2: 8238708
                CH3: 8314978
                CH4: -8388609
                CH5: -8388609
                CH6: -8388609
                CH7: -8388608
                CH8: -8388608

AFE4490 Descriptive Data:
                FrameID: 131232
                TimeTick: 2624582

AFE4490 Channel Data:
                RData: 121
                IRData: 208

pi@raspberrypi:~/inME_New $ 
```

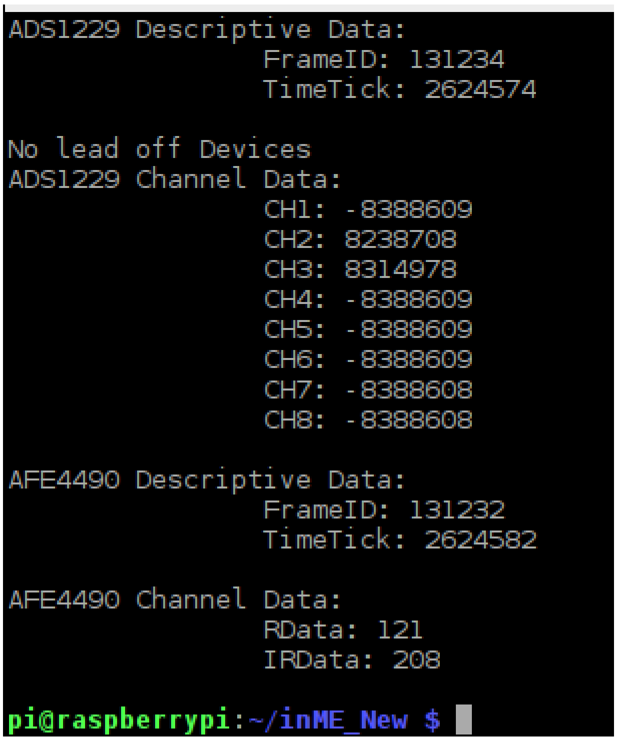
2. Data Running Horse

Code



```python
import time
from inME_API import InMEReader, ADS1229_Descriptive_DataStruct, \
    ADS1229_LeadOff_DataStruct, ADS1229_Channel_DataStruct, \
    AFE4490_Descriptive_DataStruct, AFE4490_Channel_DataStruct

inME = InMEReader()
inME.setInMEAddress("00:1A:FF:09:0A:0A")
inME.connectToInME()

while(True):
    inME.nextRecord()
    ads1229Descriptive = inME.getADS1229Descriptive()
    ads1229LeadOff = inME.getADS1229LeadOff()
    ads1229Channel = inME.getADS1229Channel()
    afe4490Descriptive = inME.getAFE4490Descriptive()
    afe4490Channel = inME.getAFE4490Channel()
    print ads1229Descriptive
    print ads1229LeadOff
    print ads1229Channel
    print afe4490Descriptive
    print afe4490Channel
    time.sleep(0.5)
except KeyboardInterrupt:
    inME.disconnectFromInME()
```

Result



```
ADS1229 Descriptive Data:
                FrameID: 131234
                TimeTick: 2624574

No lead off Devices
ADS1229 Channel Data:
                CH1: -8388609
                CH2: 8238708
                CH3: 8314978
                CH4: -8388609
                CH5: -8388609
                CH6: -8388609
                CH7: -8388608
                CH8: -8388608

AFE4490 Descriptive Data:
                FrameID: 131232
                TimeTick: 2624582

AFE4490 Channel Data:
                RData: 121
                IRData: 208

pi@raspberrypi:~/inME_New $
```

Caution

When using a loop to retrieve data endlessly from a inME Device, make sure there always be a slight pause between turns, otherwise the efficiency of the program will decrease dramatically due to insufficient data in the buffer, which cause by the data sending speed of inME. This will cause the computer busying in discarding incomplete packet.