# مقدمه

در اين گزارش با بررسي Use case هاي مربوط به فهام سعي شده است تا توابع اصلي مورد نياز براي اين Use case ها استخراج شود. اين گزارش مبناي آغاز طراحي firmware كنتور هوشمند خواهد بود. بديهي است كه در اين مرحله تمام توابع مورد نياز و نيز نحوه‌ي پياده‌سازي توابع استخراج شده كامل نخواهد بود و اين موضوع تا پايان طراحي Firmware فعال خواهد بود.

توابع ديگري نيز درآينده تعريف خواهند شد كه بر اساس الزامات قطعات استفاده شده و موارد طراحي نحوه‌ي پياده‌سازي آن توابع نيز استخراج خواهد شد. علاوه بر اين در فاز پياده‌سازي احتمالاً تعدادي از اين توابع به توابع كوچكتري شكسته خواهند شد و نيز ممكن است در مرحله‌ي پياده‌سازي به دليل محدوديت‌هاي موجود و نيز نيازهايي كه برخي از توابع بايد برآورده كنند، اين گزارش با تغييرات متعددي مواجه شود كه اين امر طبيعي مي‌باشد.

آنچه مسلم است اين گزارش مي‌تواند گام اول در طراحي firmware كنتور هوشمند و طراحي سناريوهاي مناسب براي برآورد كردن 5 use case هاي فهام باشد.

فراهم كردن قرائت‌هاي دوره‌اي كنتور

1- در اين Use case پس از دريافت يك آلارم تاريخ، كنتور عمليات فراهم كردن قرائت‌هاي دوره‌اي را آغاز مي‌كند و در صورتي كه آلارم روزانه باشد كنتور مقادير روز جاري را خوانده و در صورتي كه آلارم ماهانه باشد، مقادير مربوط به ماه گذشته را بازيابي كرده و به انضمام برچسب زماني در حافظه ذخيره مي‌كند.

2- كنتور بايد پس از دريافت درخواست قرائت دوره‌اي از DC، بسته به درخواست روزانه يا ماهانه مقادير فوق را براي DC ارسال كند. پروسه‌ي مربوط به اين Use case پس از نصب كنتور تريگر مي‌شود و ناپايان عمر كنتور به صورت پيوسته انجام مي‌شود.

3- فهرست ثبات‌هاي مورد نياز براي قرائت دوره‌اي بايد قبل از شروع فرآيند دوره‌اي در كنتور و سيستم ثبت و ذخيره شود.

4- كنتور برق بايد مقادير قرائت شده را در هر روز و زمان قابل برنامه‌ريزي ثبت كند. اگر زمان مورد نظر در كنتور ثبت نشد به صورت پيش‌فرض در ساعت 00:00:00 ثبت روزانه حادث شود.

5- كنتور برق بايد مقادير قرائت شده ماه پيش را در روز اول ماه بعد و در زمان قابل برنامه‌ريزي ثبت كند. اگر زمان مورد نظر در كنتور ثبت نشد به صورت پيش فرض در ساعت 00:00:00 روز اول هر ماه مي‌باشد.

توابع مورد نياز:

- تابع مديريت آلارم‌ها Alarm –MGNI : اين تابع وظيفه‌ي تفسير آلارم‌هاي توليد شده توسط RTC و نيز تنظيم آلارم‌هاي بعدي را بر عهده دارد و توسط تابع RTC-IRQ Handeler ( از توابع داخلي ميكرو) صدا زده مي‌شود. اين تابع در كلاس مديريت خطا، وقفه و آلارم‌ها تعريف مي‌شود.

- RTC-IRQ Handeler : اين تابع وظيفه‌ي پاسخگويي به وقفه‌هاي RTC را دارد و در كلاس RTC تعريف مي‌شود.

- تابع RTC-Set A : اين تابع وظيفه‌ي تنظيم آلارم‌هاي RTC را بر عهده دارد. اين تابع در كلاس RTC تعريف مي‌شود. در اين Use case اين تابع توسط تابع مديريت آلارم‌ها براي تنظيم آلارم‌ بعدي فراخواني مي‌شود.

- تابع Daily-Meter Reading( ) : اين تابع وظيفه‌ي قرائت روزانه را بر عهده دارد و پس از انجام عمليات قرائت روزانه مقادير خوانده شده را به انضمام بر چسب زماني در حافظه ذخيره مي‌كند. اين تابع در كلاس اندازه‌گيري تعريف مي‌شود. در اين Use case توسط تابع Alarm- MGN ( ) فراخواني مي‌شود.

- تابع Month- Metter Reading( ) : اين تابع وظيفه فراهم كردن قرائت ماهانه‌ي كنتور را بر عهده دارد و بايد مقادير قرائت شده مربوط به ماه قبل را از حافظه بازيابي كرده و حاصل آنها را به انضمام بر چسب زماني در حافظه ذخيره كند. اين تابع در كلاس اندازه‌گيري تعريف مي‌شود و در اين Use case توسط تابع Alarm- MGN ( ) فراخواني مي‌شود.

نكته 1: توابع Daily –Meter Reading ( ) و Monthly- Meter Reading ( ) بايد به گونه‌اي تعريف شوند كه بتوان مشخص كرد كه قرائت مورد نياز مربوط به كنتور گاز، آب و يا برق مي‌باشد.

نكته 2: توابع Daily –Meter Reading ( ) و Monthly- Meter Reading ( ) بايد به رجيسترهاي اندازه‌گيري و نيز تراشه‌ي اندازه‌گيري دسترسي داشته باشند.

نكته 3: توابع Daily –Meter Reading ( ) و Monthly- Meter Reading ( ) بايد به آدرس مقادير قرائت شده در حافظه دسترسي داشته باشند تا بتوانند مقادير صحيح و مورد نياز را از حافظه بازيابي كنند. (جدول آدرس شروع هر چيز)

- تابع SD-Read: اين تابع وظيفه‌ي خواندن داده‌هاي مورد نياز از حافظه جانبي را بر عهده دارد. اين تابع در كلاس ارتباط با حافظه تعريف مي‌شود.

- تابع SD- Write( ): اين تابع وظيفه‌ي نوشتن داده‌ها بر روي حافظه‌ي جانبي را بر عهده دارد. اين تابع در كلاس ارتباط با حافظه تعريف مي‌شود.

- تابع RTC-Get Time( ) : اين تابع وظيفه‌ي استخراج زمان فعلي از رجيسترهاي RTC را بر عهده دارد. اين تابع در كلاس RTC تعريف مي‌شود.

نكته 4: بايد فهرست ثبات‌هاي مورد نياز براي قرائت دوره‌اي كنتور برق و كنتورهاي متصل را به گونه‌اي كه براي توابع مربوطه قابل دسترسي باشند، از قبل تعريف شوند.

نكته 5: مقادير تعريف شده براي زمان‌هاي ثبت قرائت‌هاي دوره‌اي (موارد 3 و 4) و نيز مقادير پيش فرض- در صورت نبودن مقدار ورودي- به گونه‌اي كه براي توابع مربوط فايل دسترسي باشند، از قبل تعريف شوند. در اين Use case تابع (Alarm – MGN( ))

خلاصه توابع و كلاس‌هاي مورد نياز براي Use case1:

- كلاس مديريت خطا، وقفه و آلارم‌ها

توابع:

* Alarm –MGN ( )

- كلاس RTC

توابع:

* RTC-IRQ Handeler( )
* RTC-Set Time( )
* RTC-Get Time( )
* RTC- Set Alarm( )

- كلاس اندازه‌گيري

توابع:

* Daily- Meter Reading ( )
* Monthly –Meter Reading ( )

- كلاس ارتباط با حافظه‌ي جانبي

توابع:

* SD-Read ( )
* SD- Write ( )

نكته 6: تابع Monthly –Meter Reading ( ) در صورت نياز مي‌توان به گونه‌اي طراحي كرد كه عمل بازيابي و حاصل جمع مجموع قرائت‌هاي يك بازه زماني را به عنوان خروجي كه اين بازه مي‌تواند 1 ماه يا 30 روز و يا هر تعداد روز ديگر باشد.

**Use case 2: فراهم كردن قرائت‌هاي بنا به درخواست**

شرح اين Use case مربوط به قرائت كنتورهاي برق، آب و يا گاز بنا به درخواست مي‌باشد.

1- پس از دريافت يك دستور قرائت موردي تابع ON Demand-Meter از كلاس اندازه‌گيري فراخواني شده و آخرين مقدار قرائت شده يا هر قرائت مورد نياز ديگر به انضمام شناسه‌ي كنتور و اطلاعات مورد نياز ديگر فراهم مي‌شود.

نكته 1: تابع ON Demand-Meter Reading ( ) بايد به گونه‌اي طراحي شود كه بتوان مشخص كرد قرائت كدام يك از كنتورها مورد نياز است.

نكته 2: تابع ON Demand-Meter Reading ( )بايد به شناسه‌ي كنتور و ديگر موارد مورد نياز جهت ارائه به همراه قرائت مركز دسترسي داشته باشد. به موارد ديگر در فهام اشاره‌ي موردي نشده است و بايد بر اساس نيازهاي آتي در تابع تعريف شود و مشخص شود كه دقيقاً چه مواردي نياز است تا در پياده‌سازي تابع لحاظ شود.

2- در صورتي كه قرائت مربوط به كنتورهاي گاز و يا آب مربوط به بيش از 7 روز قبل بود بايد يك پيغام خطا ثبت كرده و به مركز ارسال كند. براي اين منظور تابع ER-REG( ) از كلاس مديريت خطا وقفه و آلارم‌ها بايد فراخواني شده و خطاي مربوط ثبت و به مركز ارسال شود.

نكته 3: در اين Use case شروع فرآيند با يك دستور از خارج از كنتور بود كه براي اين منظور نياز به يك كلاس تفسير دستورات با توابع مورد نياز آن كلاس مي‌باشد. اين كلاس توسط تيم DLMS ارائه مي‌شود.

نكته 4: خطاي توليد شده بايد مطابق با يك استاندارد تعريف شود تا براي ديگر ادوات شبكه هوشمند و DC و CAS قابل شناسايي باشد كه اين استاندارد نيز توسط تيم DLMS تعريف خواهد شد.

نكته 5: براي ارسال داده‌هاي قرائت شده بايد توابع مربوط كه توسط تيم DLMS طراحي شده فراخواني شوند.

نكته 6: در صورتي كه داده‌هايي كه قرار است قرائت شوند در حافظه‌ي جانبي باشند در تابع تابع ON Demand-Meter Reading ( ) از تابع SD-Read( ) از كلاس ارتباط باحافظه‌ي جانبي استفاده مي‌‌شود.

نكته 7: در صورتي كه نياز باشد خطاهاي توليد شده در حافظه نيز ثبت شوند در تابع ER-REG از تابع SD-Write( ) از كلاس ارتباط با حافظه‌‌ي جانبي استفاده مي‌شود.

خلاصه‌ي توابع و كلاس‌هاي مورد نياز براي Use case

- كلاس اندازه‌گيري

توابع:

* On Demand – Metter Reading ( )

- كلاس مديريت خطا، وقفه و آلارم‌ها

توابع:

* ER-REG( )

- كلاس ارتباط با حافظه‌ي جانبي

توابع:

* SD-Read ( )
* SD- Write ( )

**Use case3: فراهم كردن اطلاعات از طريق واسط ارتباط محلي مشترك**

در اين Use case پس از دريافت يك آلارم تاريخ، كنتور اطلاعات از پيش تعيين شده‌اي را از طريق واسط ارتباط محلي به تجهيزات سمت مشترك ارسال مي‌كند.

1- تابع Alarm –MGN( ) از كلاس مديريت خطا، وقفه و آلارم‌ها، آلارم توليد شده را تفسير كرده و سپس تابع مربوط به ارسال به سمت مشترك را فراخواني مي‌كند.

2- تابع Send-To IHD( ) توسط تابع Alarm –MGN( ) فراخواني شده و اطلاعات تعيين شده را به سمت تجهيزات سمت مشترك ارسال مي‌كند. اين تابع در كلاس

نكته 1: اطلاعاتي كه بايد از طريق تابع Send-To IHD( ) به سمت مشترك ارسال شوند شامل موارد زير مي‌باشند.

* آخرين مقادير قرائت شده از كنتور برق آب و گاز
* توان الكتريكي مصرفي مشترك
* موقعيت رله قطع و وصل
* نشانگر تعرفه
* شناسه‌ي تجهيز براي كنتورهاي برق، آب و گاز
* حد آستانه‌ي واقعي جريان برق
* موقعيت واقعي سوييچ شير گاز

نكته 2: تابع Send-TOIHD ( ) بايد به گونه‌اي پياده‌سازي شود كه به مقادير ذكر شده در نكته‌ي 2 دسترسي داشته باشد.

نكته 3: در صورتي كه مقادير ذكر شده در نكته 1 در حافظه ذخيره شده باشد بايد با استفاده از تابع SD-Read ( ) و مقادير مورد نظر طبق آدرس آنها در حافظه از حافظه بازيابي كند.

نكته 4: ارسال اطلاعات فوق بايد در فواصل نهايي منظم انجام پذيرد كه اين فواصل بايد از قبل در كنتور تعريف شده باشند.

خلاصه‌ي توابع و كلاس‌هاي مورد نياز براي Use Case

كلاس

توابع:

* Send-TOIHD

كلاس ارتباط با حافظه‌ي جانبي

توابع:

* SD-Read ( )

كلاس مديريت خطا، وقفه و آلارم‌ها

توابع:

* تابع Alarm Table- Update
* تابع Alarm –MGN( )

3- دوره‌هاي ارسال اطلاعات بايد قابل برنامه‌ريزي باشد براي اين منظور تابع Alarm Tab Xe-Update از كلاس مديريت خطا، وقفه و آلارم‌ها بايد به گونه‌اي پياده‌سازي شود كه كاربر بتواند به وسيله‌ي آن جدول آلارمرهاي موجود در كنتور را به روز مديريت كند.

نكته 5: براي مديريت و تفسير آلارم‌ها بايد جدولي شامل تمام آلارم‌هاي موجود در كنتور و دوره‌ي آنها و نيز اقدامات بعدي آنها موجود باشد تا تابع Alarm-MGN( ) بتواند آلارم‌هاي دريافتي را تفسير كرده و نيز آلارم‌هاي بعدي را تنظيم كند.

**Use Case 4: فراهم كردن پروفيل بار**

اين Use case مربوط به فرآيند فراهم كردن پروفيل بار مربوط ب كنتورهاي برق، آب و گاز مي‌باشد. اين فرآيند بايد پس از آغاز در سر تا سر طول عمر كنتور فعال باشد.

تنظيمات مربوط به پروفيل بار بايد فايل برنامه‌ريزي باشد.

1- پس از دريافت يك آلارم تاريخ، تابع Alarm-MGN ( ) از كلاس مديريت خطا، وقفه و آلارم‌ها پس از تفسير آلارم مربوط تابع Load-Profile را فراخواني مي‌كند.

2- تابع Load-Profile ( ) وظيفه‌ي فراهم كردن پروفيل بار از طريق خواندن مقدار رجيسترهاي اندازه‌گيري را بر عهده دارد. اين تابع در كلاس اندازه‌گيري تعريف مي‌شود.

نكته 1: تابع Load- Profile( ) بايد به پارامترهاي شناسه‌ي تجهيز و نيز جدول آلارم‌ها دسترسي داشته باشد تا در صورت نياز بتواند از آنها استفاده كند.

نكته 2: تابع Load- Profile( ) بايد به گونه‌اي پياده‌سازي شود كه در آن بتوان مشخص كرد كه پروفيل بار كدام يك از كنتورهاي برق، آب و يا گاز مورد نياز است و خروجي اين تابع نيز بايد به گونه‌اي باشد كه مشخص شود خروجي مربوط به كدام كنتور است.

نكته 3: پروفيل بار بايد حداقل شامل انرژي اكتيو (Import, Export) و انرژي راكتيو (Import, Export) براي تمام كانال‌ها باشد.

3- در صورت نياز به تغيير در بازه‌هاي ثبت پروفيل بار بايد بتوان از طريق تابع Alarm Table-Update( ) اين بازه‌ها را برنامه‌ريزي كرد.

توابع و كلاس‌هاي مورد نياز براي Use case

- كلاس مديريت وقفه، خطا و آلارم‌ها

توابع:

* تابع Alarm-MGN( )
* تابع Alarm Table-Update ( )

- كلاس اندازه‌گيري

توابع:

* Load-Profile ( )

**Use case 5 : فراهم كردن اطلاعات مربوط به كيفيت توان**

موضوع اين Use case فراهم كردن اطلاعات كيفيت توان از قبيل افت ولتاژ، افزايش ولتاژ و ولتاژ متوسط است.

در صورتي كه CAS نياز به اطلاعات مربوط به كيفت توان داشته باشد. از طريق قرائت بنا به درخواست (موضوع Use Case2) به اين اطلاعات دسترسي پيدا مي‌كند.

1- پس از دريافت يك وقفه از قسمت اندازه‌گيري (تراشه‌ي اندازه‌گيري) تابع INI-MGN ( ) كه وظيفه‌ي مديريت وقفه‌ها را بر عهده دارد، با فراخواني تابع RTC-Get Time( ) از كلاس RTC برچسب زماني وقوع وقفه را خوانده و سپس با استفاده از منابع SD- Write( ) وقفه‌ي رخ داده به انضمام برچسب زماني و ديگر اطلاعات مورد نياز را در آدرس مناسبي از حافظه ذخيره مي‌كند. اين تابع در كلاس مديريت خطا، وقفه و آلارم‌ها تعريف مي‌شود.

2- در مواردي كه نياز به ولتاژ متوسط است، اين فرآيند از طريق يك آلارم تاريخ شروع شده و توسط يك آلارم ديگر پايان مي‌پذيرد. در مورد بازه‌هاي بين اين دو زمان نيز از آلارم‌هاي ساعت استفاده مي‌شود كه همه‌ي اين موارد توسط تابع Alarm-MGN( ) مديريت مي‌شوند.

نكته 1: در موارد فوق نياز به قرائت ولتاژ، افت ولتاژ و افزايش ولتاژ مي‌باشد كه توابع مناسب جهت اين امر بايد پس از بررسي امكانات تراشه‌ي اندازه‌گيري تعريف و پياده‌سازي شوند و در اين مرحله نمي‌توان در مورد اين توابع و محل پياده‌سازي آن‌ها نظر قطعي داد.

نكته 2: بازه‌هاي زماني جهت اندازه‌گيري ولتاژ متوسط بايد قابل برنامه‌ريزي باشند كه اين امر توسط تابع Alarm Table-Update ( ) و Alarm-MGN ( ) انجام مي‌پذيرد.

خلاصه توابع و كلاس‌هاي مورد نياز براي Use case 5

- كلاس مديريت خطا، وقفه و آلارم‌ها

توابع:

* INT- MGN( )
* Alarm-MGN( )
* Alarm Table- Update ( )

-كلاس ارتباط با حافظه جانبي

توابع:

* SD-Write ( )

كلاس RTC

توابع:

* RTC-Get Time ( )
* RTC-Set Alarm( )

- كلاس مديريت اتصال متشرك

توابع مربوط به مانيتورينگ اطلاعات كيفيت توان

**Use Case 6 : فراهم كردن اطلاعات وقفه??????????????**

در اين Use case اطلاعات مربوط به وقفه‌هاي رخ داده در اتصال مشترك ثبت و به مركز و تجهيزات سمت مشترك ارسال مي‌شوند. اين وقفه‌ها به دو نوع بلند مدت (>T) و كوتاه مدت (<T) تقسيم مي‌شوند. T قابل برنامه‌ريزي است.

1- پس از وقوع يك وقفه در اتصال مشترك تابع Meter INT-MGN( ) وقفه‌‌ي رخ داده را بررسي كرده (كوتاه مدت ، بلند مدت) و به انضمام برچسب زماني در حافظه‌ي جانبي ذخيره مي‌كند. پس از آن اين اطلاعات را به سمت مشترك و مركز ارسال مي‌كند. اين تابع در كلاس اندازه‌گيري تعريف مي‌شود.

نكته 1: در اين Use case منظور از وقفه، قطعي برق و يا اختلالات رخ داده دراتصال مشترك مي‌باشد.

نكته 2: براي ارسال اطلاعات به سمت مركز بايد از توابع تعريف شده توسط تيم DLMS استفاده شود.

نكته 3: بايد يك تابع به نام Meter-Init ( ) پياده‌سازي شود تا در آغاز كار كنتور تمام متغيرهاي كنتور را تنظيم كند. اين تابع به ديگر توابعي كه مسئوليت تنظيمات اوليه كنتور را دارند مانند Alarm Table –Update دسترسي دارد.

خلاصه‌ي توابع و كلاس‌هاي موردنياز براي Use case

- كلاس اندازه‌گيري

توابع:

- Meter INT-MGN( )

- كلاس RTC

توابع:

- RTC-Get Time ( )

- كلاس ارتباط با حافظه جانبي

توابع:

- SD-Write ( )

- كلاس و ...

توابع:

- Send –TOIHD( )

- كلاس راه‌اندازي اوليه

توابع:

- Meter-Init ( )

- كلاس DLMS

توابع مورد نياز

**Use case 7: شناسايي دستكاري**

در اين Use case فرآيند شناساييو ثبت هرگونه تلاش براي دستكاري كنتور شرح داده شده است.

1- پس از دريافت يك وقفه‌ي دستكاري تابع Tamper-Detection( ) با شناسايي منبع وقفه رخ داده، رخداد مربوط را به انضمام برچسب زماني رخداد در حافظه ثبت مي‌كند تا در موارد مورد نياز بتواند اطلاعات دستكاري را در اختيار CAS قرار دهد. اين تابع در كلاس مديريت خطا، وقفه و آلارم‌ها تعريف مي‌شود.

نكته 1: دستكاري كنتور شامل دستكاري نرم‌افزار Firmware كنتور، ميدان‌هاي مغناطيسي بزرگ، باز كردن قابل اصلي و ترمينال سيم‌بندي و تلاش براي دسترسي با گذر واژه‌ي نادرست مي‌شود.

نكته 2: هر يك از مواردي كه در نكته 1 بعنوان دستكاري كنتور به آنها اشاره شد، منبع يك وقفه مي‌باشد كه آن وقفه تابع Tamper –Detection ( ) را با شناسه‌ي منحصر به فرد فراخواني مي‌كنند.

نكته 3: مديريت هر يك از موارد فوق بايد به اطلاعات مندرج در شرح اين Use case در تابع Tamper –Detection ( ) پياده‌سازي شود.

نكته 4: براي ثبت بر چسب نهايي از تابع RTC-Get Timer( ) و براي ذخيره‌ي رخدادها در حافظه از تابع SD-Writer استفاده مي‌شود.

توابع و كلاس‌هاي مورد نياز براي Use case 7

- كلاس مديريت خطا، وقفه و آلارم‌ها

توابع:

* Tamper-Detection ( )

- كلاس RTC

توابع:

* RTC-Get Time

- كلاس ارتباط با حافظه‌ي جانبي

توابع:

* SD-Writer ( )

**Use case 8: قطع يا وصل برق**

اين Use case فرآيند قطع و وصل برق شرح داده شده است. اين فرآيند مي‌تواند پس از دريافت يك دستور از CAS به صورت آي و يا در يك تاريخ از پيش تعيين شده انجام شود.

1- پس از دريافت دستور از CAS و يا دريافت يك آلارم تاريخ تابع Relay-MGN( ) كه وظيفه‌ي مديريت رله قطع و وصل را بر عهده دارد عمليات قطع و يا وصل برق را انجام مي‌دهد. سپس موقعيت رله، قرائت كنتور را به انضمام برچسب زماني در حافظه ذخيره مي‌كند. در صورتي عمليات وصل باشد بايد اطلاعات فوق را در تجهيزات سمت مشترك نيز نمايش دهد. در پايان در صورتي كه عمليات با موفقيت انجام پذيرفت بايد Logging information ها را در حافظه ذخيره كند و در غير اينصورت يك خطاي مناسب توليد كند. اين تابع در كلاس مديريت اتصال مشترك

نكته 1: براي ذخيره در حافظه از تابع SD-Write ( )، براي خواندن زمان از تابع RTC-Get Time ( )، براي ارسال به سمت مشترك از تابع Send- ToIHD ( ) و براي ثبت خطا از تابع ER-REG( ) استفاده مي‌شود.

نكته 2: تابع Alarm-MGN ( ) تابع Relogo-MGN( ) را فراخواني مي‌كند و در صورتي كه عمليات توسط يك دستور از CAS شروع شود، توابع نيم DLMS اين تابع را فراخواني مي‌كنند.

نكته 3: تابع ER- REG( ) بايد به گونه‌اي پياده‌سازي شود كه با توجه به شناسه‌ي خطاي رخ دارده عمليات مناسب را انجام دهد.

توابع و كلاس‌هاي مورد نياز براي Use case8

- كلاس مديريت خطا، وقفه و آلارم‌ها

توابع:

* Alarm-MGN( )
* ER-REG ( )

- كلاس مديريت اتصال مشترك

توابع:

* Relay- MGN( )

- كلاس ارتباط با حافظه

توابع:

* SD-Write ( )
* SD-Read( )

- كلاس RTC

توابع:

* RTC\_Get Time( )
* RTC-Set Alarm

- كلاس

توابع:

* Send-TOIHD

**Use case 9: اعمال حد آستانه برق و مديريت بار???????????????**

در اين Use case فرآيند مربوط به اعمال حد آستانه‌ي مصرف برق و فعالسازي يا غير فعالسازي آن شرح داده مي‌شود.

1- كنتور پس از دريافت دستور اعمال حد آستانه از طريق تابع Thre shold –MGN ( ) حد آستانه‌ي دريافتي را اعمال مي‌كند و عمليات را فعال مي‌كند. اين تابع در كلاس مديريت اتصال مشترك پياده‌سازي مي‌شود.

نكته 1: تابع Thre shold –MGN ( ) بايد به گونه‌اي پياده‌سازي شود كه بتواند دو نوع حد آستانه يكي براي مواقع عادي و ديگري براي مواقع اضطراري را ثبت كند و در صورتي كه حد آستانه اضطراري فعال باشد بايد اولويت را به اين مورد اختصاص دهد.

نكته 2: اين فرآيند تا زماني كه دستور غير فعالسازي درياف نشود فعال مي‌ماند و در صورت دريافت اين دستور تابع Thre shold –MGN ( ) حد آستانه را به مقدار بيشينه‌ي آن تغيير مي‌دهد.

نكته 3: در صورتي كه مشترك از حد آستانه تجاوز كرد Use case فراخواني مي‌شود.

نكته 4: موارد مربوط به شرايط مختلف قطع و وصل مشترك كه در Use case 9 شرح داده شده است بايد به صورت كامل در تابع Thre shold –MGN ( ) پياده‌سازي شود.

نكته 5: Logging information هاي مربوط به اين فرآيند مطابق با خواست فهام بايد توسط تابع SD-Write( ) درحافظه ذخيره شود.

نكته 6: در صورت امكان در زمان پياده‌سازي مي‌توان تابع Thre shold –MGN ( ) را به چند تابع كوچكتر تقسيم كرد.

نكته 7: تابع Thre shold –MGN ( ) توسط توابع تيم DLMS براي a فعال ‌سازي حد آستانه به همراه شناسه متناسب فراخواني مي‌شود.

نكته 8: پارامترهاي متغيري طبق خواست فهام در تعريف اين Use case وجود دارد كه بايد تابع Thre shold –MGN ( ) به گونه‌اي پياده‌سازي شود كه اين پارامترها قابل برنامه‌ريزي توسط CAS باشند.

توابع و كلاس‌هاي مورد نياز Use case

- كلاس مديريت اتصال مشترك

توابع:

* Thre shold –MGN ( )

- كلاس ارتباط با حافظه‌ي جانبي

توابع:

* SD-Read ( )
* SD-Write( )

- كليه توابع و كلاس‌هاي مربوط به Use case

نكته 9: توابع مربوط به Use case بايد به گونه‌اي پياده‌سازي شوند كه مشخص شود قطع و وصل شدن كنتور ناشي از فرآيند كدام يك از Use case هاي 8 يا 9 بوده است.

**Use case 10: ارسال پيام‌هاي طولاني به تجهيز سمت مشترك**

در اين Use case فرآيند ارسال پيام‌هاي طولاني براي تجهيزي سمت مشترك شرح داده شده است.

1- پس از دريافت يك پيام تابع Send- TOIHD ( ) از كلاس فراخواني شده و پيام مربوط در تجهيز سمت مشترك نمايش داده مي‌شود.

نكته 1: تابع Send- TOIHD ( ) بايد به گونه‌اي پياده‌سازي شود كه بتوان در آن نوع پيام را مشخص كرد.

نكته 2: در صورتي كه در ارسال پيام خطايي رخ دهد بايد توابع ER-MGN( ) با شناسه‌ي مناسب فراخواني شود.

نكته 3: اين Use caseتوسط دستور خارجي فعال‌سازي مي‌شود به همين سبب تابع Send- TOIHD ( ) در اينجا بايد توسط تابع نيم DLMS فراخواني شود.

نكته 4: روند اين Use caseبايد مطابق با خواست فهام در تابع Send- TOIHD ( ) براي پيام‌هاي از اين دست (هر پيام بايد شناسه‌ي مربوط به نوع خود را داشته باشد) پياده‌سازي شود.

نكته 5: در پياده‌سازي اين تابع طبق خواست فهام بايد پيام‌هاي در صف به مدت 1 ساعت منتظر بمانند كه در اين صورت به توابع RTC-Scf Alarm ( ) و Alarm-MGN( ) نيز نياز است.

خلاصه‌ي توابع و كلاس هاي مورد نياز براي Use case 10

- كلاس

توابع:

* Send- TOIHD ( )

- كلاس مديريت خطا، وقفه و آلارم‌ها

توابع:

* ER-MGN( )
* Alarm –MGN( )

- كلاس RTC

توابع:

* RTC-Set Alarm( )

- كلاس ارتباط با حافظه‌ي جانبي

توابع:

* SD-Write( )

**Use case 11 : تغيير تعرفه برق**

در اين Use case فرآيند اعمال تعرفه جديد براي كنتور برق شرح داده شده است.

1- پس از دريافت دستور مبني بر دريافت جدول جديد تعرفه‌ها، تابع Tariff Table-Update ( ) فراخواني شده و جدول تعرفه‌هاي جديد دريافت و ذخيره مي‌شود. اين تابع در كلاس اندازه‌گيري پياده‌سازي مي‌شود.

2- در صورتي كه زمان از پيش تعيين شده‌اي براي اعمال جدول تعرفه‌هاي جديد در نظر گرفته شده باشد بايد به وسيله‌ي تابع RTC-Set Alarm ( ) و از طريق تابع Alarm-MGN( ) زمان مورد نظر ثبت شود تا در وقت مقرر به وسيله تابع Tariff-Start( ) جدول تعرفه‌هاي جديد اعمال شود و در غير اينصورت در ساعت00:00 روز بعد تابع Tariff-Start( ) فراخواني مي‌شود. اين تابع در كلاس اندازه‌گيري پياده‌سازي مي‌شود.

نكته 1: براي ذخيره جدول تعرفه‌ها بايد از تابع SD-Write( )استفاده شود.

نكته 2: تابع Tariff-Start( ) بايد به گونه‌اي پياده‌سازي شود كه به جدول تعرفه‌هاي جديد و نيز جدول تعرفه‌هاي پيش فرض دسترسي داشته باشد.

3- تابع Tariff-Start( ) بايد قبل از اعمال جدول جديد تعرفه‌ها، تابع Check-Synchronization ( ) را فراخواني كند و در صورتي كه همزماني بين كنتور و DC برقرار بود جدول جديد را اعمال كرده و رويداد متناسب را ثبت نمايد و در غير اينصورت تابع ER-MGN( ) را با شناسه‌ي خطاي مناسب فراخواني كرده و جدول پيش فرض را اعمال كند.

نكته 3: در صورتي كه زماني كه براي اعمال جدول به كنتور اعمال مي‌شود مربوط به گذشته باشد بايد تابع ER-MGN ( ) با شناسه مناسب فراخواني شود.

خلاصه‌ي توابع و كلاس‌هاي مورد نياز Use Case II

- كلاس اندازه‌گيري

توابع:

* Tariff Table-Update ( )
* Tariff-Start( )

- كلاس مديريت خطا، وقفه و آلارم‌ها

توابع:

* Alarm-MGN( )
* ER-MGN( )

- كلاس RTC

توابع:

* Check-Synchronization ( )
* RTC-Set Alarm( )

- كلاس ارتباط با حافظه جانبي

توابع:

* SD-Read( )
* SD-Write( )

**Use case 12 : همزمان سازي**

در اين Use case فرآيند همزمان‌سازي ساعت دروني كنتورهاي برق، آب و گاز شرح داده شده است.

1- پس از دريافت دستور همزمان‌سازي و دريافت زمان واقعي تابع RTC-Sync ( ) فراخواني مي‌شود. اين تابع در ابتدا تابع RTC-Get Time ( ) را فراخواني كرده و زمان RTC را ثبت مي‌كند. سپس از طريق توابع RTC-Set Time ( ) زمان دريافتي را در RTC تنظيم كرده و انحراف زماني را محاسبه مي‌كند و در صورتي كه اين انحراف بيشتر از S (S از قبل قابل تعريف و تغيير است) بيشتر باشد تابع ER-MGN( ) را با شناسه متناسب فراخواني مي‌كند. تابع RTC-Sync( ) در كلاس RTC تعريف مي‌شود.

نكته 1: تابع RTC-Sync( ) بايد به گونه‌اي تعريف و پياده‌سازي شود كه در صورت لزوم بتوان با استفاده از اين تابع تمام موارد فوق را براي كنتورهاي آب و گاز نيز انجام داد.

نكته 2: اين Use case توسط يك دستور خارجي فعال مي‌شود كه بايد توسط توابع نيم DLMS آغاز شود.

خلاصه‌ي توابع و كلاس‌هاي مورد نياز براي Use case 12

- كلاس مديرت خطا، وقفه و آلارم‌ها

توابع:

* ER-MGN( )

- كلاس RTC

توابع:

* RTC-Sync( )
* RTC-Get Time ( )
* RTC-Set Time ( )

**Use case 13: ارتقاي Firmware**

در اين Use case فرآيند ارتقاي Firmware كنتور شرح داده شده است.

1- پس از دريافت دستور دريافت Firmware( ) بايد نسخه‌ي جديد دريافت شده و در حافظه ذخيره شود و صنعت آن بررسي شود و در صورت وجود تناقض نسخه‌ي دريافتي از حافظه پاك شده و خطاي متناسب توليد شود.

نكته 1: موارد ذكر شده در 1 توسط توابع نيم DLMS با فراخواني توابع SD-Delet، SD-Writer( ) و ER-REG( ) انجام مي‌پذيرد.

2- در صورتي صنعت نسخه‌ي دريافتي تأييد شده (يك دستور مبني بر اين موضوع از خارج دريافت مي‌شود) در صورتي كه زماني براي نصب اين نسخه در نظر گرفته شده باشد از طريق تابع Alarm-MGN ( ) و با استفاده از تابع RTC-Set Alarm ( ) زمان مورد نظر جهت آلارم تنظيم مي‌شود و در غير اينصورت تابع Firmware-Update ( ) از كلاس مديريت Firmware فراخواني مي‌شود. موارد فوق در تابع New Firmware-MGN( ) پياده‌سازي مي‌شود. اين تابع در كلاس مديريت Firmware تعريف مي‌شود.

3- تابع Firmware- Update( ) نسخه‌ي جديد را نصب كرده و رويداد مربوطه را به انضمام برچسب زماني با استفاده از توابع RTC-Get Time ( ) و SD –Write ( ) ثبت مي‌كند و پس از آن تابع Self-cheking( ) از كلاس مديريت خطا، وقفه و آلارم‌ها را فراخواني كرده و نتايج حاصل را به انضمام برچسب زماني ثبت مي‌كند. خلاصه توابع و كلاس‌هاي مورد نياز براي Use case 13

- كلاس مديريت firmware( )

توابع:

* New firmware-MGN ( )
* Firmware-Update( )

- كلاس مديريت RTC

توابع:

* RTC-Set Alarm
* RTC-Get Time( )

- كلاس مديريت خطا، وقفه و آلارم‌ها

توابع:

* Alarm-MGN( )
* ER-REG( )
* Self-Checking( )

- كلاس ارتباط با حافظه‌ي جانبي

توابع:

* SD-Read( )
* SD-Write( )
* SD-Delete( )

**Usecase14: تعمير در محل**

دراين Usecase فرآيند تعمير در محل و جايگزيني قطعات كنتور از قبيل باطري و ماژول مخابراتي شرح داده شده است.

نكته 1: تجهيزات بايد به گونه‌اي ساخته شوند كه پس از جدا شدن از Case كنتور يك وقفه توليد شود.

1- پس از دريافت يك وقفه مربوط به تعويض يك تجهيز بايد تابع Repair-MGN( ) از كلاس مديريت خطا، وقفه و آلارم‌ها فراخواني مي‌‌شود. اين تابع بايد عمليات مربوط را ثبت كرده و در ادامه تابع Self-Checking( ) را فراخواني كند.

خلاصه توابع و كلاس‌هاي مورد نياز براي Use case 14

- كلاس مديريت خطا، وقفه و آلارم‌ها

توابع:

* Repair-MGN( )
* Self-Checking( )

- كلاس ارتباط با حافظه

توابع:

* SD-Write( )

- كلاس RTC

توابع:

* RTC-Get Time ( )

**Use case 15: تنظيم تجهيزات**

در اين Use case فرآيند تنظيم كنتور شرح داده شده است، اين تنظيمات شامل موقعيت رله، ساعت دروني، جدول تعرفه و ... مي‌باشد.

نكته 1: تمام تنظيمات خواسته شده در اين use case در توابع مورد نياز Use case هاي ديگر برآورده شده است و امكان آن فراهم شده است.

**Use case 16: نصب تجهيزات**

در اين Use case فرآيند نصب فيزيكي تجهيزات شرح داده شده است.

1- كنتور بايد تابع Meter-Init( ) را پس از نصب فراخواني كند و اين تابع بايد تنظيمات اوليه را انجام داده، كنتورهاي آب و گاز متصل به خود را شناسايي و ثبت كرده، مدياي مخابراتي را ثبت كند و در نهايت تابع Self-Checking( ) را فراخواني كند.

نكته 1: تابع Meter-Init( ) بايد به گونه‌اي پياده‌سازي شود كه بتوان به وسيله‌ي آن اطلاعات مشترك را طبق خواست فهام در كنتور ثبت كرد.

خلاصه‌ي توابع و كلاس‌هاي مورد نياز براي Use case 16

- كلاس General

توابع:

* Meter-Init( )

- كلاس مديريت خطا، وقفه و آلارم‌ها

توابع:

* Self-Checking ( )

**Use case 17: Uninstall كردن تجهيزات**

در اين Use case فرآيند Uninstall شدن تجهيزات شرح داده شده است.

1- پس از دريافت دستور Uninstall تابع Metter-Uninstalling ( ) از كلاس General فراخواني شده و اين تابع Use case را فراخواني مي‌كند.

نكته 1: كنتور بايد قبل Uninstall شدن و صحبت خود را بازيابي كند (use case 18)

نكنته 2: بايد تابع Metter-Reset( ) در كلاس General تعريف شود تا در صورت نياز وضعيت كنتور را به حالت default بازگرداند. اين تابع با Reset ميكرو متفاوت است.

خلاصه‌ي توابع و كلاس‌هاي مورد نياز براي Usecase 17

- كلاس General

توابع:

* Meter-Uninstalling( )
* Meter-Reset( )

- كليه توابع و كلاس‌هاي Use case

**Use case 18: بازيابي وضعيت تجهيزات**

در اين Use case فرآيند بازيابي وضعيت تجهيزات شرح داده شده است. براي اين منظور كنتور بايد پارامترهاي عملكردي و پيكربندي و مقادير حقيقي خوانده شده خود و كنتورهاي آب و گاز متصل را فراهم كرده و در دسترس قرار دهد.

1- پس از دريافت يك دستور مبني بر بازيابي وضعيت تجهيزات تابع Metter-Retrieved از كلاس General فراخواني مي‌شود. اين تابع مقادير خوانده شده كنتورها را از طريق فراخواني Use case2 (فراهم كردن قرائت‌هاي بنا بر درخواست) فراهم كرده و در اختيار CAS قرار مي‌دهد. پس از آن پارامترهاي عملكردي كنتورهاي برق، آب و گاز را فراهم كرده و به انضمام Logging information هاي اين كنتورها در اختيار CAS قرار مي‌دهد.

نكته 1: تابع Metter-Retrieval ( ) بايد به گونه‌اي پياده‌سازي شود كه در زمان فراخواني آن بعنوان يك بازه‌ي زماني براي بازيابي اطلاعات در آن بازه مشخص كرد.

نكته 2: در صورتي كه بازه‌ي زماني مشخص شده نامعتبر باشد اين تابع بايد تابع ER-REG( ) را با شناسه متناسب با خطاي رخ داده فراخواني كند. (بايد از مقادير RTC استفاده كند)

نكته 3: تابع Metter-Retrieval( ) باي توسط توابع DLMS فراخواني شود.

توابع و كلاس‌هاي مورد نياز براي Use case 18

- كلاس General

توابع:

* Metter-Retrieval( )

- كلاس ارتباط با حافظه

توابع:

* SD-Read( )

- كلاس مديريت خطا وقفه و آلارم‌ها

توابع:

* ER-REG( )

- كلاس RTC

توابع:

* RTC-Get Time( )

- تمام توابع و كلاس‌هاي مورد نياز براي Use case2

**Use case 19: پشتيباني از فرآيند Self-check در تجهيزات**

در اين Use case فرآيند Self-check در تجهيزات شرح داده شده است.

1- پس از دريافت دستور Self-check و يا پس از وقوع يك رويداد كه نياز به اين فرآيند در ادامه‌ي آن است تابع Self- checking ( ) از كلاس General فراخواني شده و اين فرآيند آغاز مي‌شود.

نكته 1: اين تابع بايد وضعيت حافظه، Firmware، وضعيت مخابراتي، وضعيت باطري، كانال مخابراتي، اتصال صحيح كنتورها را بررسي كرده و نتايج را گزارش مي‌دهد.

نكته 2: در صورتي كه در هر يك از موارد اشاره شده در نكته 1 خطايي مشاهده شود بايد تابع ER-REG( ) با شناسه خطاي متناسب فراخواني شود و در صورتي كه تمام موارد پاس شدند گزارش موفقيت فرآيند ثبت شود. تمام گزارش‌ها بايد شامل برچسب زماني و نوع كنتور باشند.

نكته 3: تابع Self-Checking( ) بايد به گونه‌اي پياده‌سازي شود كه در صورت لزوم به صورت همزمان عمليات Self-check كنتورهاي آب و گاز متصل به خود را نيز فراخواني كرده و نتايج حاصل را دريافت كند.

توابع و كلاس‌هاي مورد نياز براي Use case 19

- كلاس General

توابع:

* Self-Checking ( )

- كلاس مديريت خطا، وقفه آلارم‌ها

توابع:

* ER-REG( )

ديگر توابع مورد نياز براي اين Use case بايد در زمان پياده‌سازي تابع Self-Checking( ) بسته به نحوه پياده‌سازي اين تابع تعريف و پياده‌سازي شوند.

نكته 4: بررسي تمام موارد ذكر شده در نكته 1 فقط در صورتي انجام مي‌شود كه امكان‌پذير باشد. (شرح در فهام)

**Use case 20: بررسي توپولوژي**

اين Use case توسط DC انجام مي‌پذيرد و فقط ممكن است نياز به شناسه كنتور داشته باشد كه در قرائت‌ها تعريف شده است.

**Use case 21: چك كردن ارتباطات**

در اين Use case فرآيند چك كردن وضعيت ارتباطات مخابراتي شرح داده شده است.

1- پس از نصب كنتور و يا پس دريافت يك دستور مبني بر شناسايي كنتور توسط DC كنتور تابع Com status-Response ( ) را در پاسخ به DC فراخواني كرده تا بتواند خود را به DC معرفي كند.

2- كنتور بايد تابع Com status-Check( ) را براي بررسي ارتباطات بين خود و تجهيزات تحت مديريت خود فراخواني كرده و نتايج را در اختيار CAS قرار دهد.

نكته 1: يك استاندارد براي چك كردن وضعيت مخابراتي بين DC و كنتور و نيز براي فرمت پيام‌هاي ارسالي و نتايج بايد وجود داشته باشد.

نكته 2: اين توابع بايد به گونه‌اي توسط توابع تيم DLMS فراخواني شوند.

توابع و كلاس‌هاي مورد نياز براي Use case 22 :

- كلاس General

توابع:

* Com status –Response ( )
* Com status- Check( )

- توابع ديگر در صورت لزوم در زمان پياده‌سازي بسته به نحوه پياده‌سازي تعريف مي‌شوند.

**Use Case 22 : مديريت رويدادها و آلارم‌ها**

در اين Use case فرآيند مديريت رويدادها و آلارم‌ها شرح داده شده است. رويدادها شامل خطا نيز مي‌شوند. از آنجايي كه تمام مواردي كه منجر به وقوع يك خطا مي‌شوند يا به عنوان يك رويداد بايد ثبت شوند در Use caseهاي قبلي بيان شده‌اند. در اين Use case مورد جديدي براي بيان نمي‌باشد و همه چيز بايد در تابع ER-REG ( ) تعريف و پياده‌سازي شود.

نكته 1: براي هر يك از خطاها و آلارم‌هاي مندرج در فهام و Use case 22 بايد يك كد خطاي مجزا تعريف شود به گونه‌اي كه اين كدها براي تمام تجهيزات قابل فهم باشند.