

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Dokumentácia
Dekóder QR kódov - QRlib

1 marca 2011

Michal Kučiš
xkucis00@stud.fit.vutbr.cz

pod vedením doc. Adama Herouta
herout@fit.vutbr.cz



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Tato Práce vznikla za podpory projektu MŠMT CZ.1.07/2.3.00/09.0067
TeamIT - Budování konkurenceschopných výzkumných týmu pro IT

Abstrakt

V tejto dokumentácii je popísaná knižnica QRlib. V prvej časti sú popísané všetky súbory knižnice QRlib. Medzi súbormi/triedami sú relatívne komplikované vzťahy, preto nasledujúca časť popisuje vzťahy medzi triedami v knižnici QRlib. V ďalšej časti je popísané použitie knižnice QRlib na jednoduchých príkladoch. V nasledujúcej časti je uvedený krátky popis programu QRstats, ktorý prezentuje využitie knižnice QRlib ako aj informuje užívateľa o zaujímavých informáciach obsiahnutých v QR kóde. Keďže knižnica QRlib je stále vo vývoji, v poslednej časti je uvedený popis troch najdôležitejších problémov, ktoré by mali byť doimplementované v nasledujúcich verziách knižnice.

Všetky teoretické ako aj praktické informácie boli čerpané z normy ISO/IEC 18004:2000[1]. Niektoré časti dokumentácie sú uzko späté s normou, preto sa dokumentácia odkazuje na určité strany v norme.

Obsah

1	Popis QRlib	2
1.1	base	2
1.2	BCHDecoder	2
1.3	BMP	2
1.4	BMPimg	2
1.5	CodewordBitstream	2
1.6	CodewordDecoder	3
1.7	ErrorCharacteristic	3
1.8	ErrorCharacteristicFactory	3
1.9	Exception	3
1.10	Factory	3
1.11	FuntionPatterns	4
1.12	Matrix	4
1.13	QRCode	4
1.14	QRDataDecoder	4
1.15	RSDecoder	4
1.16	StaticData	4
1.17	TableForAlphanumMode	5
1.18	Vec2n	5
2	Vzťahy medzi triedami v QRlib	6
3	Použitie QRlib	7
3.1	Inicializácia knižnice a načítanie QR kódu	8
3.2	Výpis informácií o QR kóde	8
3.3	Výpis dát uložených v QR kóde	8
3.4	Modifikácia QR kódu	9
4	QRstats	9
4.1	Vstup	9
4.2	Výstup	9
4.2.1	Chyba pri načítaní interných dát	9

4.2.2	Všetko prebehlo v poriadku	9
4.3	Chyby	11
4.3.1	Fatálne chyby	11
4.3.2	Závažné chyby	12
4.3.3	Informačné chyby	12
5	To-Do	13
5.1	Reed-Solomon kódovanie	13
5.2	Bose-Chaudhuri-Hocquenghem kódovanie	14
5.3	Podpora ďalších módov	14

1 Popis QRlib

Popis uvedený v tejto časti sa vzťahuje na zdrojové súbory *.h a *.cpp. Väčšina súborov deklaruje/definuje len jednu triedu rovnakého názvu ako je samotná trieda.

1.1 base

Súbor obsahuje deklarácie základných typov, definuje užitočné základné makrá. V tomto súbore sú definované funkcie InitLib()/DestroyLib(), ktorú je nutné zavolať pred/po použití ostatných častí knižnice QRlib. InitLib() inicializuje statické dáta, DestroyLib() globálne uvoľňuje naalokovanú pamäť rôznymi časťami QRlib. Pozor, DestroyLib() neslúži ako garbage collector, funkcia len dealokuje pamäť, ktorú nie je možné uvoľniť inými mechanizmami.

1.2 BCHDecoder

BCHDecoder umožňuje dekódovať a odstraňovať chyby v dátach, ktoré sú zakódované pomocou BCH kódu. Aktuálna verzia neimplementuje opravu chýb, len extrahuje dáta. Pre ďalšie podrobnosti viz stať TODO/Bose-Chaudhuri-Hocquenghem kódovanie.

1.3 BMP

BMP slúži ako Facade pattern nad triedou BMPimg.

1.4 BMPimg

BMPimg umožňuje načítavanie súborov typu BMP. Trieda bola prevzatá zo zdroju na internete. Definícia triedy bola typu OpenSource, voľne šíriteľná a modifikovateľná.

1.5 CodewordBitstream

CodewordBitstream umožňuje vyberať z matice bity Codeword-ov. CodewordBitstream sa využíva nasledovne:

1. Inicializuje sa objekt typu QRCode
2. Vytvorí sa inštancia CodewordBitstream. Konštruktoru sa nastaví požadovaný objekt typu QRCode, ktorý sa má spracovať.

3. Pomocou funkcie `GetCodewordAndMoveToNext` alebo pomocou funkcie `GetBit()` sa získavajú dáta z časti pre codewordy. Funkcia `GetCodewordAndMoveToNext()` vráti 8 bitov a posunie sa v súbore na ďalších 8 bitov. Pri ďalšom zavolaní funkcie `GetCodewordAndMoveToNext()` funkcia vráti nasledujúcich 8 bitov. Funkcia `GetBit()` vráti bit ale neposunie sa v súbore na ďalší bit. Po opätovnom zavolaní funkcie `GetBit()` funkcia vráti hodnotu toho istého bitu. Pre posunutie na ďalší bit sa použije funkcia `NextBit()`. Počet bitov je možné získať zo štruktúry `ErrorCharacteristic`. Objekt `CodewordBitstream` očakáva, že behom svojej existencie sa `QRCode` nebude meniť.
4. Zruší sa objekt typu `CodewordBitstream`.
5. Objekt typu `QRCode` sa môže meniť.

1.6 CodewordDecoder

`CodewordDecoder` obaľuje dekódovanie demaskovaných dát z QR kódu. Hlavná metóda `CodewordDecoder::Decode()` očakáva na vstupe pole bitov, ktoré sú boli získané z QR kódu a boli následne odmaskované. Poradie bitov je špecifikované v [1] na strane 46. Výstup funkcie tvorí pole bitov v takom poradí, aby bolo možné na výstupe priamo dekódovať Mód, Character Count Indicator, prípadne ďalšie bity. Vstup tvoria codewordy, tak ako je vyobrazené v [1] na strane 46. Codeword D1 sa nachádza na 0. až 7. bite, Codeword D14 sa nachádza na 8. až 15. pozícii, atď. Výstup tvorí pole bitov, tak ako je popísané v [1] na strane 21. Samotná trieda taktiež sa snaží opravovať chyby, volaním príslušných metód triedy `RSDecoder`.

1.7 ErrorCharacteristic

`ErrorCharacteristic` je štruktúra, ktorá umožňuje uchovávať informácie o počte a type codewordov. Taktiež uchováva informácie o spôsobe kódovania. Štruktúra kopíruje riadok tabuliek uvedených v [1] na stranách 35-44.

1.8 ErrorCharacteristicFactory

Trieda umožňuje vytvoriť štruktúru, ktorá bude obsahom kopírovať tabulky uvedené v [1] na stranách 35-44. Trieda pri vytváraní poľa `ErrorCharacteristic` otvorí súbor obsahujúci potrebné dáta, spracuje ich a naplní týmito dátami výstupné pole.

1.9 Exception

Súbor obsahuje definície základných výnimiek, ktoré knižnica `QRlib` môže vyvolať pri svojom behu.

1.10 Factory

Továrenská trieda, ktorá umožňuje vytvárať objekt typu `StaticData`. Vytváranie objektu typu `StaticData` je časovo náročné, preto je `StaticData` navrhnutý ako Singleton.

1.11 FuntionPatterns

Trieda obaľuje `Matrix<bool>`, ktoré majú tvar a obsah kopírujúci tvar `Position` a `AlignmentPattern` v QR kódach. Function patterny sú reprezentované booleovskou maticou, kde čierny bod v QR kóde je reprezentovaný hodnotou `true`. Biely bod je reprezentovaný hodnotou `false`.

1.12 Matrix

Šablona `Matrix` spravuje matice. Šablona `Matrix` používa triedu `MatrixData`, ktorá umožňuje zdieľanie dát medzi viacerými inštanciami triedy `Matrix`. Toto správanie je vhodné v tom prípade, keď pracujeme s väčšou maticou a požadujeme napr. submaticu tejto matice. V tomto prípade sa dáta matice nekopírujú do novej submatice, ale submatica bude zdieľať dáta zdrojovej matice. Submatica bude mať obmedzený prístup do dát len na oblasť, ktorá submatici prináleží.

Ak je nutné vytvoriť kópiu dát matice, je nutné použiť metódu `Matrix::GetDeepCopy()`, ktorá vytvorí kópiu matice a aj dát. Ostatné metódy definované v `Matrix` pracujú so sdieľanými prvkami matice (pracujú so shallow copy matice).

1.13 QRCode

Trieda `QRCode` dedí z triedy `Matrix<bool>` všetky jej vlastnosti a metódy. Zároveň rozširuje funkcionality triedy o vlastnosti a metódy, ktoré prináležia výhradne QR kódom (neplatia všeobecne pre matice). Táto trieda reprezentuje QR kód pomocou booleovskej matice, kde čierny bod v QR kóde je reprezentovaný v `QRCode` hodnotou `true`. Biely bod v QR kóde je v triede `QRCode` reprezentovaný ako `false`.

1.14 QRDataDecoder

Trieda `QRDataDecoder` dekoduje `QRCode` a vracia dáta, ktoré sú QR kóde uložené. Vstup je matica uvedená v [1] na strane 5, výstup tvorí reťazec znakov. Dáta kódované v QR kóde v Móde 8-Bit Byte môžu obsahovať znaky s hodnotou `0x00`. Tieto znaky je možné uložiť do reťazca typu `std::string`. Tieto znaky (a znaky, ktoré sa nachádzajú za nimi) nie je možné správne interpretovať pomocou bežných C-čkových funkcií. Preto je vhodné používať na zobrazovanie výsledkov reťazce typu `std::string`.

Táto trieda neimplementuje spracovávanie všetkých módov, pre podrobnosti viz. stať `TODO/Podpora ďalších módov`.

1.15 RSDecoder

`RSDecoder` umožňuje dekódovať a odstraňovať chyby v dátach, ktoré sú zakódované pomocou Reed-Solomon kódu. Aktuálna verzia neimplementuje opravu chýb, len extrahuje dáta. Pre ďalšie podrobnosti viz stať `TODO/Bose-Chaudhuri-Hocquenghem kódovanie`.

1.16 StaticData

`StaticData` informácie, ktoré nezávisia na konkrétnom QR kóde. Trieda umožňuje:

- overuje existenciu verziu QR kódu

- vracia počet modulov na stranu pre zadanú verziu QR kódu
- vracia pre danú verziu počet modulov na stranu
- uchováva informácie o pozícii Alignment patternov pre všetky verzie QR kódov
- uchováva masky QR kódov
- vracia ErrorCharacteristic pre danú verziu a pre daný Error Correction Level

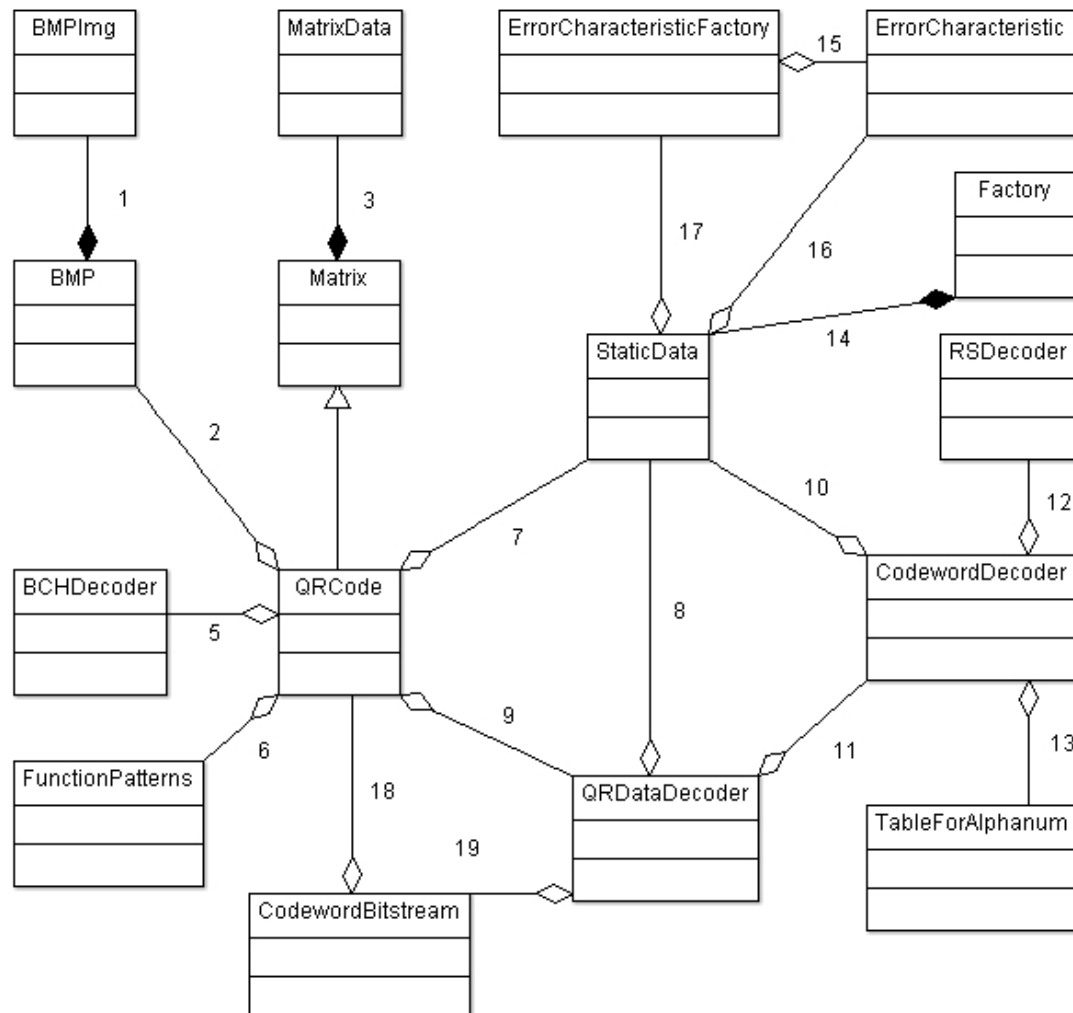
1.17 TableForAlphanumMode

Trieda slúži na dekódovanie znakov pri móde Alphanumeric do znakov v ASCII. V [1] na strane 21 je uvedená tabuľka prevodu.

1.18 Vec2n

Vec2n implementuje 2 prvkové vektory celých čísel. Tento vektor sa používa pri určovaní 2D súradníc resp. rozmerov 2D objektov.

2 Vzťahy medzi triedami v QRlib



Poznámka: Predchádzajúci diagram slúži ako pomôcka na vysvetlenie vzťahov v rámci knižnice QRLib. Diagram nezobrazuje presne asociácie a kompozície, tak ako sú fyzicky napísané v zdrojových kódach knižnice QRLib.

1. BMP upravuje rozhranie BMPImg pre jednoduchšie využitie QRlib-om. BMP tvorí Facade Pattern nad BMPImg.
2. QRCode využíva triedu BMP na načítavanie dát zo súborov typu bmp.
3. MatrixData uchováva dáta matice. Pre prístup k týmto dátam sa používa trieda Matrix. Tento návrh umožňuje zdieľať dáta matice medzi viacerými maticami.
4. QRCode je odvodená z triedy Matrix. Trieda QRCode rozširuje rozhranie Matrix o ďalšie metódy, ktoré sú charakteristické pre QR kódy.

5. BCHDecoder sa v QRCode používa na dekodovanie Format a Version Information modulov.
6. QRCode umožňuje vypisovať počet modulov s nesprávnou hodnotou. Na zisťovanie počtu chýb sa používa porovnávanie matíc. Pri porovnávaní sa porovná časť matice QR kódu s maticou, ktorú vráti FunctionPatterns.
7. StaticData poskytuje rozhranie, ktoré umožňuje overovať vlastnosti QR kódu. Trieda QRCode sama na sebe testuje tieto vlastnosti a cez svoje rozhranie poskytuje výsledky užívateľovi.
8. QRDataDecoder dekoduje dáta uložené v QR kóde. Pre správne dekodovanie potrebuje vedieť určité dáta, ktoré poskytuje trieda StaticData.
9. QRCode dekoduje svoje dáta z dátových codewordov pomocou triedy QRDataDecoder.
10. Dáta v QR kóde sú uložené do blokov. Bloky sú v rámci codewordov navzájom preskladávané. CodewordDecoder umožňuje codewordy preskladať tak, aby dáta bolo možné dáta uložené v QR kóde sekvenčne prechádzať a dekodovať. Pre svoj správny beh potrebuje dáta uložené v StaticData.
11. QRDataDecoder pri dekodovaní dát v QR kóde preskladáva bloky codewordov pomocou triedy CodewordDecoder.
12. CodewordDecoder preskladáva dáta a opravuje chyby, ktoré sa vyskytli v QR kóde. Na opravu chýb používa triedu RSDecoder.
13. TableForAlphanum uchováva tabuľku prevodu znakov pri Alphanumeric móde. Tento prevod využíva trieda CodewordDecoder pri dekodovaní dát z QR kódu.
14. StaticData majú privátny konštruktor. Ak je nutné kdekoľvek v programe použiť inštanciu StaticData, je nutné túto inštanciu vytvoriť pomocou triedy Factory.
15. ErrorCharacteristic je inicializované pomocou dát uložených v externom súbore. ErrorCharacteristic je štruktúra, ktorú naplňa ErrorCharacteristicFactory.
16. Ak je nutné prísť k dátam uložených v ErrorCharacteristic, je nutné použiť triedu StaticData pre prístup k týmto dátam.
17. StaticData naplňa dáta ErrorCharacteristic pomocou triedy ErrorCharacteristicFactory.
18. CodewordBitstream poskytuje prístup k bitom v dátových a chybových codewordoch.
19. Rozhranie CodewordBitstream využíva trieda QRDataDecoder na prístup k bitom codewordov.

3 Použitie QRlib

Táto časť má slúžiť ako jednoduchý návod na používanie knižnice.

3.1 Inicializácia knižnice a načítanie QR kódu

Najjednoduchšie použitie QRLib ilustruje nasledujúci kód:

```
InitLib ("."); // potrebná inicializácia knižnice QRLib

// načíta bitmapu obsahujúcu QR kód
// bitmapa obsahuje moduly široké a vysoké 9 pixelov
// funkciu informujeme o šírke modulu 2. parametrom
QRCode qrcode;
qrcode.LoadFromBMPFile ("QRCode.bmp", 9);

if (qrcode.IsValid()) // ak je načítaný kód v poriadku, tak...
{
    // nejak spracujeme QR kód
}

DestroyLib (); // po dobre vykonanej práci, upraceme a končíme
```

3.2 Výpis informácií o QR kóde

Keď požadujeme výpis niektorých vlastností QR kódu, je možné použiť nasledujúci kód:

```
// predpokladáme, že qrcode je premenná typu QR kód
// qrcode je správne inicializovaná a obsahuje dáta platného QR kódu
std::cout << "Version=" << qrcode.GetVersionFromNumModules () << std::endl;
std::cout << "Mask type=" << qrcode.GetNumMaskPatternLT () << std::endl;

// Zobrazíme si maticu po odmaskovaní:
std::cout << endl << "Matrix:" << endl;
qrcode.PrintBitsDemasked (qrcode.GetNumMaskPatternLT());
```

3.3 Výpis dát uložených v QR kóde

Nasledujúci kód získava dáta uložené v QR kóde a vypisuje ich na štandardný výstup.

```
// predpokladáme, že qrcode je premenná typu QR kód
// qrcode je správne inicializovaná a obsahuje dáta platného QR kódu

std::string data;
QRDataDecoder decoder;
decoder.Decode (qrcode, qrcode.GetErrorLevelLT(),
               qrcode.GetNumMaskPatternLT(), data);
```

3.4 Modifikácia QR kódu

Ak uložený QR kód horizontálne zrkadlený (pravá strana je vymenená za ľavú), je nutné pred dekódovaním QR kód správne otočiť:

```
// predpokladáme, že qrcode je premenná typu QR kód
std::cout << "Before:" << endl;
qrcode.PrintToStdout();
std::cout << endl;

// najprv zrkadlovo obrátíme QR kód
// os zrkadlenia je priamka rovnobežná s x-ovou osou
qrcode = qrcode.GetVerticalFlippedMatrix ();

// potom QR kód správne dotočíme:
for (int i = 0; i < 2; i++)
    qrcode = qrcode.GetRotated ();

std::cout << "After:" << endl;
qrcode.PrintToStdout();
```

4 QRstats

Tento projekt slúži ako ukážka použitia QRlib a aj ako nástroj, pomocou ktorého je možné analyzovať QR kódy uložené v bitmape.

4.1 Vstup

Program očakáva ako parameter programu cestu k bitmape, ktorá obsahuje QR kód. Aby program správne načítal QR kód, je nutné, aby modul v QR kóde bol 10 pixelov široký. Toto správanie je možné zmeniť zmenou hodnoty konštanty `c_sizeOfModul`.

4.2 Výstup

4.2.1 Chyba pri načítaní interných dát

Keď program vypíše chybu:

```
ERROR: .\centersOfAlignPatterns.dat cannot be found
```

je nutné skopírovať obsah QRdata do pracovného adresára QRstats.exe.

4.2.2 Všetko prebehlo v poriadku

Ak program spracuje bezchybný súbor, výstup môže vyzeráť nasledovne:

```
Loading file '../input/ok/001-decode.bmp'
    načítava dáta qr kódu....
Position patterns (differces j= 4): 0x0 22x0 0x22
    našiel position patterny (akceptuje 4 chyby v patterne)

Position patterns are valid. Let's go decode...
    pomocou position patternov je možné zistiť pozíciu QR kódu
Found QRcode from 0x0 to 29x29
    QR kód sa nachádza na zadaných pozíciach
Total number of modules: 29x29
    počet modulov, ktoré sa nachádzajú v nájdenom QR kóde
Version: 3      verzia QR kódu, ktorá je získaná z rozmerov QR kódu
Invalid position patterns: 0/147
    koľko/z koľkých modulov position patternov nemá správnu hodnotu
Invalid position patterns with separators: 0/192
    koľko modulov position patternov vrátane bielych okrajov position patternov,
    ktoré sa nachádzajú v QR kóde nemá správnu hodnotu
Invalid alignment patterns: 0/25
Invalid timing patterns: 0/26
Invalid function patterns with separators: 0/243
    koľko funkčných modulov vrátane bielych okrajov position patternov nemá
    správnu hodnotu
Error level LT: Q
    error correction level získaný z ľavých horných (left top) format information modulov
Error level RT/LB: Q
    error correction level získaný z pravých horných a ľavých dolných format information
    modulov
Mask pattern LT: 1      hodnota mask patternu získaná ľavých horných format information
    modulov
Mask pattern RT/LB: 1 Version from version info RT: NONE (Version<7)      výpis infor-
    muje, že version information moduly vo verziách <7 nie sú prítomné
    ak by QR kód bol verzie >= 7, za dvojbodkou by sa nachádzala verzia
Version from version info LB: NONE (Version<7) Decoded data (maskLT='1'; errlvlLT='Q')
    na dekodovanie sa použije format information moduly umiestnené vľavo hore
DATA='http://www.fit.vutbr.cz/ herout/'
    a pomocou vyššie uvedených hodnôt masky a error level sa dekodovali tieto dáta

...
```

...

Brute force? Let's try...

výpis informuje, že dekóder sa pokúša nájsť QR kód metódou Brute force
skúša všetky možné otočenia/veľkosti/pozície vo vstupnom súbore
a vyberie tú variantu, ktorá dávala najmenšiu percentuálnu chybu na pozíciách
function patternov
je možné vypnúť hľadanie pomocou hrubej sily nastavením konštanty
c_isBruteForceEnabled

ďalší výpis je totožný s výstupom uvedeným skôr:

Found QRcode from 0x0 to 28x28
Total number of modules: 29x29
Version: 3
Invalid position patterns: 0/147
Invalid position patterns with separators: 0/192
Invalid alignment patterns: 0/25
Invalid timing patterns: 0/26
Invalid function patterns with separators: 0/243
Error level LT: Q
Error level RT/LB: Q
Mask pattern LT: 1
Mask pattern RT/LB: 1
Version from version info RT: NONE (Version<7)
Version from version info LB: NONE (Version<7)
Decoded data (maskLT='1'; errlvlLT='Q')
DATA='http://www.fit.vutbr.cz/ herout/'

4.3 Chyby

Program vykoná viacero kontrol, pokiaľ kontrola neprebehla v poriadku vypíše správu s prefixom "PROBLEM: ". Pokiaľ chyba nie je závažná, program pokračuje ďalej. V tejto časti budú vypísané chybové správy, ktoré kontroly môžu vypisovať.

4.3.1 Fatálne chyby

Do tejto kategórie spadajú chyby, ktoré spôsobia ukončenie behu aplikácie.

- Program argument is missing
- .\centersOfAlignPatterns.dat cannot be found
- .\errorCharacteristics.dat cannot be found
- \${CESTA.SUBORU_TYPU_BMP} cannot be found
- Error is not resolved

4.3.2 Závažné chyby

Do tejto kategórie spadajú chyby, ktoré zabránia nájdeniu QR kódu bežným postupom (vyhľadáním Position patternov). Program môže vyhľadať QR kód pomocou metódy Brute Force (pokiaľ je nastavená konštanta `c.isBruteForceEnabled` na `true`).

- Number of position patterns is invalid
- Too many position patterns are found
- No position pattern is found
- Only one position pattern is found
- Position patterns has invalid position

4.3.3 Informačné chyby

Do tejto kategórie spadajú chyby, ktoré signalizujú, že niečo nie je v poriadku. Program pokračuje vo svojom behu ďalej.

- Unsupported mode in encoded data (?error?)
- Encoded data are corrupted
- Something terrible happened during decoding data
- Length of data is zero ? Error!
- Error has occurred during decoding data
- Error level LT != Error version RT/LB
- Mask pattern LT != Mask pattern RT/LB
- Version info RT != Version info LB
- Version modules != Version info RT
- Version modules != Version info LB
- (Mask pattern LT != Mask pattern RT/LB) or (Error level LT != Error level RT/LB)
- Duplicat rows (this and prev): 1 5 7
- Duplicat columns (this and prev): 1 5 7
- Similar rows (this and prev)(differences $\neq 4$): 1 5 7
- Similar columns (this and prev)(differences $\neq 4$): 1 5 7
- White rows: 1 5 7
- Black rows: 1 5 7
- White columns: 1 5 7

- Black columns: 1 5 7
- White rows (differences $j=4$): 1 5 7
- Black rows (differences $j=4$): 1 5 7
- White columns (differences $j=4$): 1 5 7
- Black columns (differences $j=4$): 1 5 7
- White rows ignore borders: 1 5 7
- Black rows ignore borders: 1 5 7
- White columns ignore borders: 1 5 7
- Black columns ignore borders: 1 5 7
- White rows ignore borders (differences ≤ 4): 1 5 7
- Black rows ignore borders (differences ≤ 4): 1 5 7
- White columns ignore borders (differences ≤ 4): 1 5 7
- Black columns ignore borders (differences ≤ 4): 1 5 7
- Result of brute force \neq Result of recommended sequence

5 To-Do

V knižnici QRLib je nutné doprogramovať nasledujúce časti:

5.1 Reed-Solomon kódovanie

Oblasť pre "Data and Error Correction Codewords" je kódovaná Reed-Solomon kódovaním. Toto kódovanie umožňuje odstraňovať chyby, ktoré sa môžu vyskytnúť v dátových slovách. Algoritmus opravy chýb nie je implementovaný QRLib. V aktuálnej verzii sa QRLib nepokúša odstraňovať chyby, QRLib skopíruje dátové slová aj s chybami.

Algoritmus dekódovania je vhodné pridať do rsdecoder.h do RSDecoder::Decoder(..). V aktuálnej verzii je metóda volaná funkciami, ktoré očakáva dekódovanie/opravu chýb. Ak bude algoritmus implementovaný, ostatný kód bude dekódovanie okamžite používať.

1. parameter metódy obsahuje pole bitov. Toto pole obsahuje bity dátových a aj opravných slov. 2. parameter je pole bitov, ktoré bude dekóder vracať. Dekóder by mal nastaviť správnu veľkosť poľa bitov, tak aby mohol obsahovať bity všetky dátových slov. Bity opravných slov by sa tomto argumente už nemali nachádzať. 3. parameter obsahuje počet dátových a aj opravných slov. Dekóder môže očakávať, že platí $3.\text{parameter} == \text{sizeof}(1.\text{parameter}) * 8$. 4. parameter obsahuje počet dátových slov, ktoré sú v 2. parametre. Vždy by malo platiť $4.\text{parameter} == \text{sizeof}(2.\text{parameter}) * 8$.

5.2 Bose-Chaudhuri-Hocquenghem kódovanie

Oblasť "Format Information" a "Version Information" je kódovaná BSH kódovaním. Toto kódovanie umožňuje zistiť chybu, ktorá vznikla zámenov jednotlivých bitov. V aktuálnej verzii sa QRLib nepokúša odstraňovať chyby, QRLib skopíruje dátové slová aj s chybami.

Algoritmus dekódovania je vhodné pridať do bchdecoder.h do BCHDecoder::Decoder (...). V aktuálnej verzii je metóda volaná funkciami, ktoré očakáva dekódovanie/opravu chýb. Ak bude algoritmus implementovaný, ostatný kód bude dekódovanie okamžite používať.

1. parameter metódy obsahuje pole bitov. Toto pole obsahuje bity dátové a aj opravné.
2. parameter je pole bitov, ktoré bude dekóder vracať. Dekóder môže očakávať, že pole je tak veľké, aby mohli byť do poľa pridané všetky dátové bity. Opravné bity sa do tohto parametru nedávajú.
3. parameter obsahuje počet dátových a aj opravných bitov. Dekóder môže očakávať, že platí `3.parameter == sizeof(1.parameter)`.
4. parameter obsahuje počet dátových bitov, ktoré sú v 2. parametre. Vždy by malo platiť `4.parameter == sizeof(2.parameter)`.

5.3 Podpora ďalších módov

QRLib podporuje "Numeric Mode", "Alphanumeric Mode", "8-bit Byte Mode", "Mixing Mode" a "Terminator". V [1] sú popísané ďalšie módy, ktoré nie sú v aktuálnej verzii QRLib podporované. Podporu ďalších módov je možné pridať doimplementovaním dekódovania týchto módov. Toto je možné uskutočniť pridaním kódu do súboru qrdatadecoder.h do metódy QRDataDecoder::Decode (...). Spôsob by mal byť zjavný z implementácie už implementovaných módov.

Reference

- [1] *ISO/IEC 18004: Automatic identification and data capture techniques – Bar code symbology – QR Code*. International Organization for Standardization, Geneva, Switzerland, 2000, ISBN 0580364127.