**Changes to CPP**

//—

/\*\*

\* \brief Get Track’s Year

\*

\* \param[out] info buffer pointer char array to be updated with result

\*

\* Extract the Year from the current file handles track ID3 tag information.

\*

\* \warning ID3 Tag information may not be present on all source files.

\* Otherwise may result in non-sense.

\* It is possible to add it with common tools outside of this project.

\*/

void SFEMP3Shield::trackYear(char\* infobuffer){

getTrackInfo(TRACK\_YEAR, infobuffer);

}

//————

/\*\*

\* \brief Get Track’s Genre

\*

\* \param[out] info buffer pointer char array to be updated with result

\*

\* Extract the Album from the current file handles track ID3 tag information.

\*

\* \warning ID3 Tag information may not be present on all source files.

\* Otherwise may result in non-sense.

\* It is possible to add it with common tools outside of this project.

\*\

void SFEMP3Shield::trackGenre(char\* infobuffer){

getTrackGenre(TRACK\_GENRE, infobuffer);

}

void SFEMP3Shield::getTrackGenre(uint8\_offset, char\* infobuffer){

//disable interrupts

if(playing\_state == playback) {

disableRefill();

}

//record current file position

uint32\_t currentPos = track.curPosition();

//skip to end

track.seekEnd((-128+offset));

track.read

\*infobuffer=num;

//seek back to saved file position

track.seekSet(currentPos);

//renable interupt

if(playing\_state == playback) {

enableRefill()

}

}

//------------------------------------------------------------------------------

/\*\*

\* \brief Fetch ID3 Tag information

\*

\* \param[in] offset for the desired information desired.

\* \param[out] infobuffer pointer char array of filename to be read.

\*

\* Read current filehandles offset of track ID3 tag information. Then strip

\* all non readible (ascii) characters.

\*

\* \note this suspends currently playing streams and returns afterwards.

\* Restoring the file position to where it left off, before resuming.

\*/

void SFEMP3Shield::getTrackInfo(uint8\_t offset, char\* infobuffer){

//disable interupts

if(playing\_state == playback) {

disableRefill();

MP3player.pauseMusic();

}

//record current file position

uint32\_t currentPos = track.curPosition();

//skip to end

track.seekEnd((-128 + offset));

if((offset==TRACK\_ALBUM)||(offset==TRACK\_ARTIST)||(offset==TRACK\_TITLE)){

//read 30 bytes of tag informat at -128 + offset

track.read(infobuffer, 30);

infobuffer = strip\_nonalpha\_inplace(infobuffer);

}

if((offset==TRACK\_YEAR)){

//read 4 bytes of tag informat at -128 + offset

track.read(infobuffer, 4);

infobuffer = strip\_nonalpha\_inplace(infobuffer);

}

//seek back to saved file position

track.seekSet(currentPos);

//renable interupt

if(playing\_state == playback) {

enableRefill();

}

}

**Changes to Header**

#define TRACK\_GENRE 127

#define TRACK\_YEAR 93

class SFEMP3Shield {

public:

void trackYear(char\*);

void trackGenre(char\*);